



① BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 199 00 369 A 1**

⑤ Int. Cl. 7:  
**G 06 F 13/12**  
G 06 F 13/28  
G 06 F 13/16

① Aktenzeichen: 199 00 369.6  
② Anmeldetag: 7. 1. 1999  
④ Offenlegungstag: 8. 7. 1999

DE 199 00 369 A 1

③ Unionspriorität:  
09/004002 07. 01. 1998 US

⑦ Anmelder:  
National Semiconductor Corp., Santa Clara, Calif.,  
US

⑦ Vertreter:  
BOEHMERT & BOEHMERT, 80801 München

⑦ Erfinder:  
Brief, David, Medlin, IL; Fensore, David J., New  
Gloucester, Me., US; Waterson, Kent Bruce, South  
Portland, Me., US; Dean, Gregory Lewis, Standish,  
Me., US

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤ Vorrichtung und Verfahren zur Ausführung einer Steuerübertragung auf einem Universal Serial Bus

⑤ Verfahren und Vorrichtung zur Ausführung einer Steuerübertragung auf einer Universellen Seriellen Bus(USB)-Einheit. Eine USB-Einheit umfaßt einen Speicherraum zum Lesen und Schreiben von Daten, die über ein USB-Netzwerk übertragen werden. Den Speicherraum teilen sich eine Vielzahl von Endpunkten. Ein Host initiiert durch das Übertragen eines SETUP-Tokens auf einen ersten Endpunkt eine Steuerübertragung. Der Endpunkt muß den SETUP-Token akzeptieren. Wenn der erste Endpunkt den SETUP-Token nicht erwartet, oder wenn ein anderer Endpunkt aktiv ist, speichert die Einheit den Token, bis ein Puffer zugeordnet ist, und der erste Endpunkt wird aktiviert.

DE 199 00 369 A 1

Die Erfindung betrifft eine Vorrichtung und ein Verfahren zur Ausführung einer Steuerübertragung über einen Universal Serial Bus (USB), insbesondere das Antworten auf ein Setup, auf Daten und auf Statustransaktionen einer Steuerübertragung.

Der USB ist eine normierte Peripherieschnittstelle für den Anschluß von Personalcomputern an eine große Vielzahl von Geräten, beispielsweise digitale Telefonleitungen, Monitore, Modems, Mäuse, Drucker, Scanner, Spielcontroller, Tastaturen und andere Peripheriegeräte. Der USB ersetzt deshalb bekannte Schnittstellen, wie serielle Anschlüsse RS-232C, parallele Anschlüsse, PS/2-Schnittstellen und Spiel/MIDI-Anschlüsse.

Beim USB nutzen alle angeschlossenen Geräte, die mit einem Personalcomputer mit Hilfe einer einzigen Verbindungsart verbunden sind, einen "Tiered Star"-Aufbau. Ein Host-Personalcomputer umfaßt einen einzigen USB-Controller. Der Host-Controller bildet die Schnittstelle zwischen dem USB-Netzwerk und dem Host-Personalcomputer. Der Host-Controller steuert alle Zugriffe auf USB-Mittel und überwacht die Bus-Topologie. Ein USB-Hub bildet USB-Anschlußpunkte für USB-Geräte.

Ein Beispiel eines "Tiered Star"-Aufbaus eines USB-Netzwerks ist in Fig. 1 dargestellt. Der Host PC 100 ist ein typischer Personal Computer mit einer USB-Anschlußverbindung über den Host-Controller 102. Der Host-Controller 102 ist direkt mit dem Basis-Hub 110 verbunden, welcher typischerweise als Teil des Host-PCs 100 implementiert ist. Das Verbindungsgerät 120, der Hub 130 und das Gerät 140 sind mit dem Host-Controller 102 durch den Basis-Hub 110 verbunden. Die Geräte 132 und 134 sind mit dem Host-Controller 102 durch den Hub 130 und den Hub 110 verbunden.

Das Verbindungsgerät 120 umfaßt die Geräte 124, 126 und den Hub 122. Der Hub 122 des Verbindungsgerätes 120 ist mit dem Host-Controller durch den Hub 110 verbunden. Die Geräte 124 und 126 des Verbindungsgerätes 120 sind mit dem Host-Controller 102 durch den Hub 122 und den Hub 110 verbunden. Ein praktisches Beispiel für ein Verbindungsgerät ist ein integriertes Drucker- und Faxgerät. Der Drucker könnte das Gerät 124 sein, und das Faxgerät könnte das Gerät 126 sein.

Der in Fig. 1 dargestellte verbundene "Star-Aufbau" erlaubt die Datenübertragung von verschiedenen Geräten in den und aus dem Host-PC 100. Wenn Daten von dem Host an ein Gerät übermittelt werden, so ist dies ein Abwärtstransport durch die miteinander verbundenen Hubs. Wenn Daten von den Geräten an den Host übermittelt werden, ist dies ein Aufwärtstransport durch die miteinander verbundenen Hubs.

Die USB-Hubs und -Geräte können ohne einen kompletten Neustart des USB-Netzwerkes miteinander verbunden oder voneinander getrennt werden. Auf die Verbindung eines Gerätes oder eines Hubs an einen Aufwärts-Hub wird der Aufwärts-Hub den Host-Controller über eine Statusänderung informieren. Gemäß dem USB-Protokoll wird der Host-Controller den Anschluß des Hubs aktivieren, an welchen das Gerät angeschlossen wurde. Der Host-Controller wird anschließend jedem Gerät eine einmalige funktionelle Adresse zuweisen. Die Anschlüsse werden bei der Zuweisung der einmaligen funktionellen Adressen durch den Host-Controller 102 der Reihe nach aktiviert. Nach dem Anschluß eines Verbindungsgerätes weist der Host-Controller jedem Gerät in dem Verbindungsgerät eine einmalige funktionelle Adresse zu. In Fig. 1 wird den Geräten 124, 126, 132, 134 und 140 zusammen mit den Hubs 110, 122 und 130 nach der Verbindung mit das USB-Netzwerk eine einmalige funktionelle Adresse zugewiesen.

Eine USB-Funktion ist ein USB-Gerät, welches in der Lage ist, Informationen auf dem Bus zu übertragen und zu empfangen. Eine Funktion kann eine oder mehrere Konfigurationen haben, wobei jede die Schnittstellen definiert, die das Gerät bilden. Jede Schnittstelle ist ihrerseits von einem oder mehreren Endpunkten gebildet.

Ein Endpunkt ist das Datenzuflußende oder Datenabflußende. Ein Endpunktkanal dient der Bewegung der Daten zwischen USB und Speicher, und vervollständigt den Weg zwischen dem USB-Host und dem Endpunkt der Funktion. Ein USB-Gerät kann zu jeder Zeit bis zu 16 derartige Endpunktkanäle unterstützen. Jeder Endpunktkanal hat dieselbe funktionelle Adresse.

Bei der Initialisierung eines Gerätes assoziiert der Host-Controller einen Kanal mit den Endpunkt-Funktionen. Der Kanal ermöglicht es dem Host-Controller Daten in und aus einem Host-Speicherpuffer an die und von den Endpunkten zu bewegen. Auf dem USB können zwei Arten von Kanal-Kommunikationsbetriebsarten implementiert werden: Strom und Nachricht. Stromdaten haben keine definierte USB-Struktur. Nachrichtendaten haben eine definierte USB-Struktur.

Bei der Initialisierung wird einem Kanal ein Anspruch auf USB-Buszugriff und -Bandbreitennutzung zugewiesen. Diese Zuweisung bestimmt die Priorität bei der Übertragung von Daten über einen speziellen Kanal zu einem Endpunkt. Die Eigenschaften des Endpunktes werden bei der Initialisierung auch mit dem Kanal assoziiert. Derartige Eigenschaften umfassen den maximalen Datennutzlastenbereich, die Direktionalität der Übertragungen und andere geeignete, charakteristische Daten. Diesen Eigenschaften werden von dem Host bei der Datenübertragung über den Kanal benutzt.

Die Zuweisung eines Anspruchs auf USB-Buszugriff und -Bandbreite an einen Kanal, ermöglicht es dem Host-Controller, Aufwand zu betreiben, um zu sichern, daß alle Eingangs-/Ausgangsdaten-Anfragen an die Endpunkte, über die verfügbare Bandbreite des USB-Busses angemessen bedient werden können. Die Zuweisung von Ansprüchen auf Buszugriff und -bandbreite zu einem Kanal, begrenzt die Zuordnung zu später konfigurierten Geräten. Wenn die Bandbreite eines USB-Busses vollständig zugeordnet ist, kann nachfolgend konfigurierten Geräten keine Bus-Bandbreite zugeordnet werden. Hieraus folgt, daß nachfolgend konfigurierte Geräte keine Kanäle zugeordnet werden können.

Nachdem der Initialisierungsprozeß abgeschlossen ist, ist die Zuordnung der Kanäle zu speziellen Endpunkten eines Gerätes fest, und kann nicht verändert werden, bis das Gerät getrennt wird, oder ein Reset ausgeführt wird. Entsprechend wird Geräten mit einer Vielzahl von Endpunkt-Funktionen eine Vielzahl von Kanälen zugeordnet, die jeder mit einem speziellen Endpunkt verbunden sind.

Jeder Endpunkt ist eine adressierbare Einheit auf dem USB und muß auf IN- und OUT-Token (Eingangs- und Ausgangs-Token) von dem USB-Host-Controller antworten. Die IN-Token zeigen an, daß der Host den Empfang von Informationen von einem Endpunkt verlangt. Die OUT-Token zeigen an, daß der Host Informationen an einen Endpunkt senden wird.

Auf das Erkennen eines an einen Endpunkt adressierten IN-Tokens ist der Endpunkt für das Antworten mit einem Da-

tenpaket verantwortlich. Wenn der Endpunkt momentan verstopft ist, wird ein STALL-Handshake-Paket gesendet. Wenn der Endpunkt freigegeben ist, aber keine Daten anliegen, wird ein negatives Quittier-Handshake-Paket (NAK) gesendet.

In ähnlicher Weise ist der Endpunkt beim Erkennen eines an einen Endpunkt adressierten OUT-Tokens für den Empfang eines von dem Host gesendeten Datenpaketes und dessen Speicherung in einem Puffer verantwortlich. Wenn der Endpunktkanal am Ende einer Datenübertragung zeitweise blockiert ist, wird ein STALL-Handshake-Paket (Quittier-Paket) gesendet. Wenn der Endpunktkanal am Ende einer Datenübertragung zeitweise nicht freigegeben ist, so wird kein Handshake-Paket gesendet. Wenn der Endpunktkanal empfangsbereit ist, aber kein Puffer vorhanden ist, in welchem die Daten gespeichert werden können, wird ein NAK-Handshake-Paket gesendet.

Ein nicht aktivierter Endpunkt oder ein Endpunkt, welcher momentan nicht auf einen Endpunktkanal abgebildet ist, antworten nicht auf IN-, OUT- oder SETUP-Token.

Der USB definiert vier Arten der Datenübertragung über einen Kanal: Steuer-, Bulk-, Interrupt- und Isochron-Übertragung.

Steuer-Übertragungen werden von dem Host zur Konfigurierung eines Gerätes nach dem Anschluß an einen Hub genutzt. Steuerübertragungen können von dem Host-Controller auch für die Implementierung spezieller Transaktionen mit einem Gerät genutzt werden.

Bulk-Übertragungen sind sequentielle Übertragungen, im allgemeinen Übertragungen von großen Datenmengen. Bulk-Übertragungen liefern zuverlässige Transaktionen durch die Nutzung von Fehlererkennung und dem erneuten Senden von fehlerhaften Daten. Die Busbandbreite, welche für eine Bulk-Übertragung zugeordnet wird, kann jede verfügbare Bandbreite sein, da Bulk-Übertragungen nicht zeitempfindlich sind.

Interrupt-Übertragungen sind kleine spontane Datentransaktionen, die von einem Gerät ausgehen.

Isochron-Übertragungen sind kontinuierliche Echtzeitdatentransaktionen. Isochron-Übertragungen sind einem Teil einer USB-Netzwerk-Bandbreite ausschließlich zugeordnet, um einen rechtzeitigen Abschluß der Isochronen-Übertragungen zu sichern.

Die USB-Vorschrift definiert ein Steuer-Übertragungsprotokoll zur Nutzung bei der Konfigurierung, Führung und Überprüfung des Status eines Gerätes. Eine Steuerübertragung setzt sich aus einer Setup-Transaktion, welche angeforderte Informationen von dem Host zu dem Gerät bewegt, optionalen Daten-Transaktionen, welche Daten in einer durch die Setup-Transaktion angezeigte Richtung senden, und einer Status-Transaktion, welche Status-Informationen von dem Gerät zu dem Host zurücksendet, zusammen. Die Setup-Transaktion spezifiziert den Umfang der Daten, welche während der Daten-Transaktion gesendet werden sollen.

Das Auftreten einer IN- oder OUT-Daten-Transaktion (Eingangs- oder Ausgangs-Daten-Transaktionen) in der Steuer-Transaktion liefert drei mögliche Transaktionssequenzen: eine Steuer-Schreib-Sequenz, eine Steuer-Lese-Sequenz und eine Steuer-"No-Data"-Sequenz. Jede dieser Sequenzen wird im folgenden im Detail beschrieben.

In Fig. 4A ist eine Steuer-Schreib-Sequenz dargestellt. Die Steuer-Schreib-Sequenz umfaßt drei Stufen: Eine Setup-Stufe, eine Daten-Stufe und eine Status-Stufe. Die Setup-Stufe umfaßt eine SETUP-Transaktion 410 mit einem DATA0-PID. Die SETUP-Transaktion 410 legt fest, daß eine OUT-Daten-Stufe folgen wird. Die Daten-Stufe umfaßt eine OUT-Transaktion 412 mit einem DATA1-PID. Die OUT-Transaktion 412 wird gefolgt von einer OUT-Transaktion 414 mit einem DATA0-PID. Dieser Transaktion folgen so viele Transaktionen, wie benötigt werden, um die notwendigen Daten von dem Host zu übertragen. Dieses hängt vom Umfang der Daten in dem Host und vom Umfang der übertragenen Pakete ab. Der DATA-PID wechselt zwischen 1 und 0 für die in der Daten-Stufe genutzten OUT-Transaktionen. Die abschließende OUT-Transaktion 416 beendet die Daten-Stufe. Die Status-Stufe umfaßt eine einzelne IN-Transaktion 418 mit einem DATA1-PID.

Fig. 4B zeigt eine Steuer-Lese-Sequenz. Die Steuer-Daten-IN-Sequenz umfaßt drei Stufen: eine Setup-Stufe, eine Daten-Stufe und eine Status-Stufe. Die Setup-Stufe umfaßt eine SETUP-Transaktion 420 mit einem DATA0-PID. Die SETUP-Transaktion legt fest, daß ein IN-Daten-Stufe folgen wird. Die Daten-Stufe umfaßt eine IN-Transaktion 422 mit einem DATA1-PID. Auf die IN-Transaktion 422 folgt eine IN-Transaktion 424 mit einem DATA0-PID. Diesem Token folgen so viele Transaktionen, wie benötigt werden, um die notwendigen Daten von dem Gerät zu übertragen. Dieses hängt vom Umfang der Daten in dem Gerät und dem Umfang der übertragenen Pakete ab. Der DATA-PID wechselt zwischen 1 und 0 für die in der Daten-Stufe genutzten IN-Transaktionen. Die abschließende IN-Transaktion 426 beendet die Daten-Stufe. Die Status-Stufe umfaßt eine einzelne OUT-Transaktion 428 mit einem DATA1-PID.

Fig. 4C zeigt eine Steuer-"No-Data"-Sequenz. Die Steuer-"No-Data"-Sequenz umfaßt zwei Stufen: eine Setup-Stufe und eine Status-Stufe. Die Setup-Stufe umfaßt eine SETUP-Transaktion 420 mit einem DATA0-PID. Die Status-Stufe umfaßt eine einzelne IN-Transaktion 428 mit einem DATA1-PID.

In Fig. 3 ist ein Flußdiagramm einer Setup-Transaktion dargestellt. Die Transaktion beginnt mit der Sendung des SETUP-Tokens 310 von dem Host. Eine Setup-Transaktion umfaßt immer einen DATA0-PID für das Datenfeld. Das Gerät antwortet anschließend durch das Senden eines ACK-Handshakes, um die SETUP-Transaktion zu vervollständigen.

Fig. 5 zeigt ein Flußdiagramm einer IN-Transaktion. Die IN-Transaktion beginnt mit dem Senden eines IN-Tokens 510 von dem Host an das Gerät. Das Gerät sollte dann mit dem passenden DATA-Paket 512 (entweder ein DATA0 oder ein DATA1-Paket) antworten. Wenn das Gerät jedoch zeitweise nicht in der Lage ist, ein DATA-Paket zurückzusenden, so wird es statt dessen ein NAK-Handshake 514 zurücksenden. Wenn das Gerät nicht in der Lage ist, ein DATA-Paket zurückzusenden, und eine Host-Intervention für die Wiederaufnahme verlangt, wird es einen STALL-Handshake 516 zurücksenden. Auf den Empfang des DATA-Paketes 512 wird der Host mit einem ACK-Handshake 518 antworten.

Fig. 6 zeigt ein Flußdiagramm einer OUT-Transaktion. Die OUT-Transaktion beginnt mit dem Senden eines OUT-Tokens 610 von dem Host an das Gerät. Der Host sendet danach das passende DATA-Paket 612 (entweder ein DATA0- oder ein DATA1-Paket). Wenn das Gerät das DATA-Paket 612 ohne Fehler empfängt, und wenn es zum Empfang eines anderen Paketes bereit ist, sendet es ein ACK-Handshake 614 zurück. Wenn das Gerät das DATA-Paket 612 ohne Fehler empfängt, dieses Paket jedoch erneut vom Host gesendet haben möchte, sendet es ein NAK-Handshake 614 zurück. Der NAK-Handshake wird benutzt, wenn ein für ein DATA-Paket vorgesehener Endpunkt in einem Zustand ist, welcher zeitweise den Empfang des DATA-Paketes verhindert. Wenn das Gerät das DATA-Paket 612 empfängt, jedoch noch in ei-

nem STALL-Zustand ist, sendet es einen STALL-Handshake zurück, um anzuzeigen, daß der Host nicht versuchen sollte, das Paket erneut zu senden. Wenn das Datenpaket 612 mit einem CRC- oder einem Bit-Stuff-Fehler ('Verstopft-Fehler') empfangen wurde, wird kein Handshake zurückgesendet.

Der USB verlangt, daß ein angeschlossenes Gerät einen unerwarteten SETUP-Token akzeptiert.

5 Um SETUP-Token und das entsprechende DATA0-Paket zu akzeptieren, ordnen USB-Geräte Speicher ausschließlich für den Empfang dieser Daten zu. Für den Empfang von Token ordnen USB-Geräte zusätzlich wenigstens einem anderen Endpunkt Speicher zu. Da ein USB-Gerät bis zu sechzehn Endpunkte unterstützen kann, ist jedem Endpunkt Speicher für den Empfang von Token für diesen Endpunkt und für das Senden und Empfangen von Daten oft ausschließlich zugeordnet.

10 Zu jedem Zeitpunkt kann nur ein Endpunkt aktiv sein. Dementsprechend wird Speicher, der mit einem inaktiven Endpunkt verbunden ist, nicht gelesen, um Daten über den USB zu übertragen. In gleicher Weise wird der mit einem inaktiven Endpunkt verbundene Speicher nicht mit Daten von dem USB beschrieben. Trotzdem bewahrt jeder Endpunkt typischerweise ausschließlich zugeordneten Speicher.

Deshalb besteht Bedarf an einem USB-Gerät, welches keine jedem Endpunkt ausschließlich zugeordnete Speicherpuffer verlangt. Darüberhinaus besteht Bedarf an einem USB-Gerät, welches keinen ausschließlich einem Steuer-Endpunkt zugeordneten Speicher verlangt, aber in der Lage ist, unerwartete SETUP-Token zu akzeptieren.

Aufgabe der Erfindung ist es, einen Speicherraum zu schaffen, auf den eine Vielzahl von Endpunktkanälen zur Datenübertragung zugreifen kann.

20 Eine weitere Aufgabe der Erfindung ist es, einen Speicherraum zu schaffen, auf den eine Vielzahl von Endpunktkanälen für den Datenempfang zugreifen kann.

Eine weitere Aufgabe der Erfindung ist es, daß ein SETUP-Token in einem Ready-Zustand immer akzeptiert wird.

Eine weitere Aufgabe der Erfindung ist es, den Speicherumfang zu reduzieren, welcher einem Steuer-Endpunktkanal ausschließlich zugeordnet werden muß.

25 Es ist eine weitere Aufgabe der Erfindung, einen Speicherraum zu schaffen, der von verschiedenen Arten von Endpunktkanälen gemeinsam genutzt werden kann.

Es ist eine weitere Aufgabe der Erfindung, einem Datenprozessor die Überprüfung eines Setup-Paketes einer Steuerübertragung in einen Speicher zu erlauben, und dem Prozessor anschließend die Vorbereitung des Resets der Steuerübertragung zu erlauben.

30 Es ist eine weitere Aufgabe der Erfindung, einen Controller zu schaffen, welcher die Steuerübertragung ausführt und welcher automatisch mit einem geeigneten Handshake antwortet.

Es ist Aufgabe der Erfindung, einen Doppelpufferspeicher zum Senden und Empfangen von Daten zu schaffen.

Es ist eine weitere Aufgabe der Erfindung, eine Speicherleitung für die Übertragung von Daten von einem USB zu einem Speicher zu schaffen.

35 Gemäß einem Aspekt der Erfindung umfaßt ein Verfahren zum Betreiben eines Steuerendpunktes auf einem USB die folgenden Schritte:

- Bereitstellen eines Sender-Empfängers, welcher zum Senden und Empfangen von Daten betreibbar mit dem USB verbunden ist;
- Bereitstellen eines Controllers, welcher mit dem Sender-Empfänger funktionell verbunden ist, wobei der Controller Daten an den Sender-Empfänger zur Übertragung über den USB liefert, und wobei der Controller Daten annimmt, welche der Sender-Empfänger über den USB empfängt;
- Bereitstellen eines Speichers, welcher betreibbar mit dem Controller verbunden ist, wobei der Controller Daten in dem ersten Speicher sichern kann und Daten aus dem ersten Speicher kopieren kann;
- Bereitstellen einer Funktion, welche funktionell mit dem Speicher und dem Controller verbunden ist, wobei die Funktion Befehle an den Controller liefern kann, Daten in dem Speicher sichern kann und Daten aus dem Speicher kopieren kann;
- Empfangen eines SETUP-Tokens von einem Host an dem USB, wobei der SETUP-Token anzeigt, daß der Host eine Steuerübertragung ausführen wird;
- Empfangen eines ersten DATA0-Paketes von dem Host auf dem USB, wobei das erste DATA0-Paket anzeigt, daß die Steuerübertragung einer Datenstufe mit wenigstens einer Datentransaktion umfassen wird; und
- Zuordnen eines Teiles des Speichers für die wenigstens eine Datentransaktion, wobei die Funktion die Zuordnung ausführt und eine erste Speicheradresse an den Controller liefert, und wobei die erste Speicheradresse auf eine erste Speicherstelle innerhalb des Teiles des Speichers zeigt.

55 Gemäß einem anderen Aspekt der Erfindung umfaßt ein Gerät zur Verbindung mit einem USB:

- einen Pufferspeicher zur Speicherung von Daten;
- einen Controller, welcher funktionell mit dem Pufferspeicher und einem USB verbunden ist, der Controller aufweisend:
  - einen ersten und einen zweiten Speicheradressen-Zeiger, wobei der erste und der zweite Speicheradressen-Zeiger eine Adresse innerhalb des Pufferspeichers halten;
  - Logik für einen Endpunktkanal-Zustand zur Bewahrung des gegenwärtigen Zustandes eines Endpunktkanals, wobei die Logik anzeigt, ob der Endpunktkanal Daten zur Übertragung über den USB enthält;
  - eine Funktion, welche funktionell mit dem Controller und dem Pufferspeicher verbunden ist, wobei die Funktion Daten in dem Pufferspeicher speichert und die entsprechende Adresse dem ersten und zweiten Pufferspeicher-Zeiger zuweist, wobei der Controller automatisch ein NAK sendet, wenn der Controller einen IN-Token einer Steuerlese-Übertragung empfängt, und wenn die Logik des Endpunktkanal-Zustandes anzeigt, daß der Endpunktkanal keine Daten enthält, und wobei der Controller automatisch die Daten sendet, wenn die Logik des Endpunktkanals

Zustandes anzeigt, daß der Endpunktkanal keine Daten enthält.

Diese und andere Aufgaben, Merkmale und Vorteile sind in der folgenden Beschreibung und den zugehörigen Zeichnungen offenbart.

**Fig. 1** zeigt ein Blockdiagramm eines USB-Netzwerkes.

**Fig. 2** zeigt ein Blockdiagramm einer bevorzugten Ausführungsform eines erfindungsgemäßen Schnittstellenmoduls für ein USB-Gerät.

**Fig. 3** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während einer SETUP-Transaktion.

**Fig. 4A** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während einer Steuer-Schreib-Übertragung.

**Fig. 4B** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während einer Steuer-Lese-Übertragung.

**Fig. 4C** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während eines Steuer-"No-Data"-Übertragung.

**Fig. 5** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während einer IN-Transaktion.

**Fig. 6** zeigt ein Flußdiagramm des Datenflusses über ein USB-Netzwerk während einer OUT-Transaktion.

**Fig. 7** zeigt ein Blockdiagramm einer bevorzugten Ausführungsform eines erfindungsgemäßen Schnittstellenmoduls für ein USB-Gerät.

**Fig. 8** zeigt ein Blockdiagramm einer bevorzugten Ausführungsform eines Endpunkt-Controllers aus **Fig. 7**.

**Fig. 9** zeigt ein Blockdiagramm einer bevorzugten Ausführungsform eines erfindungsgemäßen Programmiermodells für ein USB-Gerät.

**Fig. 10** zeigt ein Flußdiagramm der Zustände und Zustandsübergänge eines Endpunktes während einer erfindungsgemäßen Steuertransaktion.

Die in dieser Anmeldung beschriebene Erfindung ist Bestandteil einer größeren Anzahl von Erfindungen, die in weiteren Patentanmeldungen desselben Anmelders offenbart sind, und die mittels Referenz zum Bestandteil der vorliegenden Anmeldung gemacht werden. Es handelt sich um die folgenden Patentanmeldungen: (i) US-Patentanmeldung Nr. 09/003,963 mit dem Titel "Apparatus and Method of Implementing a Versatile USB Endpoint Pipe", (ii) US-Patentanmeldung Nr. 09/004,003 mit dem Titel "An Apparatus and Method of Implementing a USB Endpoint Pipe with Double Buffering Support", (iii) US-Patentanmeldung Nr. 09/004,004 mit dem Titel "An Apparatus and Method for Transmitting and Receiving Data Into and Out of a Universal Serial Bus Device", (iv) US-Patentanmeldung Nr. 09/003,897 mit dem Titel "Apparatus and Method of Transmitting and Receiving USB Isochronous Data", (v) US-Patentanmeldung Nr. 09/004,005 mit dem Titel "An Apparatus and Method for Providing an Interface to a Compound Universal Serial Bus Controller", und (vi) US-Patentanmeldung Nr. 09/003,925 mit dem Titel "An Apparatus and Method for Initializing a Universal Serial Bus Device".

Darüberhinaus ist das Dokument "USB Node Interface Architecture Specification" (Version 0.6) der National Semiconductor Corporation als Anhang A Bestandteil der vorliegenden Anmeldung.

Im folgenden wird unter Bezugnahme auf **Fig. 2** ein erfindungsgemäßes USB-Gerät **210** beschrieben. Die bevorzugte Ausführungsform des Gerätes erfüllt die USB-Vorschrift, Version 1.0, wobei sie eine Anzahl von Vorteilen gegenüber herkömmlichen USB-Geräten aufweist. Das USB-Gerät **210** umfaßt ein USB-Schnittstellenmodul **220**, eine Registergruppe **214**, einen RAM **212** und eine Bus-Schnittstelleneinheit (BIU) **216**.

Das Schnittstellenmodul **220** ist mit einem USB-Netzwerk zum Senden und Empfangen von Daten verbunden. Das Schnittstellenmodul **220** ist auch mit der Registergruppe **214**, dem RAM **212** und der BIU **216** verbunden. Der Kernbus **218** ermöglicht es dem Schnittstellenmodul **220**, auf die Registergruppe **214** und den RAM **212** zuzugreifen.

Die BIU **216** liefert den Zugang zum Adreßraum des Kernbusses **218** mittels der Ausführung von Lese- und Schreibübertragungen als ein Master und des Registerzugriffs als ein Slave. Zusätzlich stellt die BIU eine Datenleitung zur Verfügung. Die Datenleitung kann dazu genutzt werden, Daten auf ihrem Weg zu einer Speicherstelle zeitweise zu halten.

In **Fig. 7** ist das USB-Schnittstellenmodul **220** aus **Fig. 1** beschrieben. Das USB-Schnittstellenmodul **220** liefert die Schnittstelle, um ein USB-Gerät **210** mit einem USB-Netzwerk zu verbinden. Das Schnittstellenmodul **220** umfaßt einen integrierten USB-Sender-Empfänger **714**, eine serielle Schnittstellenmaschine (SIE) (**712**) und den Endpunkt-Controller **710**.

Der Sender-Empfänger **714** liefert die physikalische Schnittstelle mit einem USB-Netzwerk. Der Sender-Empfänger umfaßt zwei Treiber und Empfänger, welche die physikalischen Schichtvorschriften der Version 1.0 der USB-Vorschriften erfüllen.

Die serielle Schnittstellenmaschine **712** ist betreibbar mit dem Sender-Empfänger **714** verbunden und weist ein physikalisches (PHY)-Niveau und ein Medienzugriffs-Steuer (MAC)-Niveau auf. Das physikalische Niveau umfaßt einen digitalen Zeittakt-Wiederaufnahmeschaltkreis, einen digitalen Pannfilter, einen End\_Of\_Packet-Erkennungsschaltkreis und einen Bit-Stuff- und Bit-Destuff-Logikschaltkreis auf.

Das Medienzugriffs-Steuerniveau umfaßt ein Paket zur Formatierung, zur Erzeugung und zur Überprüfung der zyklischen Redundanzüberprüfung, und zur Erkennung der Endpunktadresse, und liefert die notwendige Steuerung, für das Geben der durch den Endpunkt-Controller **202** für den spezifizierten Endpunktkanal bestimmten NAK-, ACK- und STALL-Antworten. Die serielle Schnittstellenmaschine **712** ist auch dafür verantwortlich, daß USB-spezifische Ereignisse, wie ein Reset, Suspend und Resume erkannt und übermittelt werden.

Der Endpunkt-Controller **710** ist betreibbar mit der seriellen Schnittstellenmaschine **712** verbunden und liefert die Schnittstelle für die USB-Funktionsendpunkte.

Der Endpunkt-Controller **710** arbeitet als ein Master/Slave auf dem Kernbus **118** und unterstützt alle USB-Endpunkt-Klassen: Steuerung, Bulk, Isochron und Interrupt. Der Slave-Zugriff wird benutzt, um auf interne Steuerregister der Registergruppe **214** zuzugreifen. Der Master-Zugriff wird benutzt, um Daten zwischen Endpunkten und einem RAM **212** zu übertragen.

Der Endpunkt-Controller **710** bewahrt separate Zustandsinformation für jeden Endpunktkanal. Zustandsinformation,

die für unmittelbare Entscheidungen verlangt wird, wie z. B. zum Antworten mit einem NAK- oder STALL-Handshake, wird in dem Endpunkt-Controller 710 zu jeder Zeit für alle Endpunktkanäle in einem teilweisen Endpunktkanal-Speicher gehalten. Diese Information wird benutzt, um die Medienzugriffs-Steuerschicht der seriellen Schnittstellenmaschine 712 beim Geben der passenden ACK-, NAK- und STALL-Antwort zu führen. Andere für den Endpunkt-Betrieb notwendige Information wird in einer Endpunkt-Anordnung des RAMs 212 gehalten.

Unter Bezugnahme auf Fig. 8 wird die Struktur des Endpunkt-Controllers 710 im Detail beschrieben. Der Endpunkt-Controller 710 umfaßt einige funktionelle Blöcke. Insbesondere umfaßt der Endpunkt-Controller 710 eine Medienzugriffs-Steuerschnittstelle 802, einen Gerätefunktionsblock 804, eine Endpunkt-Zustandsmaschine 806, einen Endpunktkanal-Teilspeicher (PEPS) 808, einen Adressgenerator 810 und ein Daten-Ausrichtungsmultiplexer 812.

Die Medienzugriffs-Steuerschnittstelle 802 ist mit der Endpunkt-Zustandsmaschine 806 und dem Daten-Ausrichtungsmultiplexer 812 verbunden, um für das Medienzugriffs-Steuerniveau der seriellen Schnittstellenmaschine 712 eine Schnittstelle zu liefern. Einige Bereiche des Endpunkt-Controllers 710 arbeiten mit einem Taktgeber auf dem Kernbus 218. Andere Bereiche arbeiten mit einem lokalen Taktgeber des Endpunkt-Controllers 710. Die Medienzugriffs-Steuerschnittstelle 802 synchronisiert die Datenbytes zwischen den verschiedenen Bereichen mit einem asynchronen Handshake für jedes aus dem Endpunkt-Controller 710 übertragene Byte.

Der Gerätefunktionsblock 804 führt Gerätefunktionen aus, wie die Adressüberprüfung und -erzeugung. Der Gerätefunktionsblock ist betreibbar mit der Endpunkt-Zustandsmaschine 806 verbunden, um Informationen in Bezug auf den Endpunktkanal zu liefern, auf welchen gegenwärtig zugegriffen wird.

Die Endpunkt-Zustandsmaschine 806 führt Funktionen bezüglich der Bewahrung des Zustandes eines aktiven Endpunktkanals aus. Die Zustandsmaschine 806 bewahrt den Zustand eines aktiven Endpunktkanals mit definierten Übergängen, um Hardware- und Software-Wechselwirkungen zu synchronisieren. Der gegenwärtige Zustand ist für einen Host-Controller über ein Endpunkt-Zustandsregister sichtbar. Alle Endpunktkanäle teilen sich deshalb die Endpunkt-Zustandsmaschine 806.

Der Endpunktkanal-Teilspeicher 808 speichert zu jeder Zeit für jeden Endpunktkanal Zustandsinformationen, welche für unverzügliche Entscheidungen, wie das Antworten mit einem NAK- oder STALL-Handshake verlangt wird.

Der Adressgenerator 810 liefert die Adressen, die für das Lesen und Schreiben in den RAM über den Kernbus notwendig sind.

Der Daten-Ausrichtungsmultiplexer 812 stellt die vor der Übertragung auf dem Kernbus notwendigen Datenbyte-Ausrichtungen zur Verfügung.

Eine bevorzugte Ausführungsform der erwähnten Endpunktkanal-Anordnung wird im folgenden im Detail beschrieben. In Fig. 9 ist ein bevorzugtes Programmierungsmodell gezeigt. Das Programmierungsmodell zeigt eine Vielzahl von Registern 910, welche dazu genutzt werden, die Operationen auf einem USB-Gerät zu steuern und zu überwachen. Das Modell umfaßt weiterhin eine in dem Speicher 920 angeordnete Endpunktkanal-Anordnung 922. Die Endpunktkanal-Anordnung 922 wird benutzt, um die für eine bestimmte Endpunkt-Übertragung spezifischen Endpunkt-Parameter zu steuern.

Das Programmierungsmodell gemäß Fig. 6 umfaßt eine Registergruppe 910 und einen Speicher 920. Teile der Registergruppe 910 und Teile des Speichers 920 sind aufeinander abgestimmt, um eine spezielle Endpunkt-Übertragung zu überwachen und zu steuern. In Fig. 9 ist auch eine bevorzugte Ausführungsform der Beziehung zwischen der Registergruppe 910 und dem Speicher 920 dargestellt.

Die Endpunktkanal-Anordnung 922 umfaßt für jeden der sechzehn Endpunktkanäle ein erstes Steuerwort CW0 und ein zweites Steuerwort CW1. Die Endpunktkanal-Anordnung 922 ist in dem Speicher auf einer Adresse angeordnet, die in einem Endpunkt-Anordnungs-Basisadressregister (EABAR) 912 gespeichert ist. Diese Endpunktkanal-Anordnung ist mittels einer vier-Bit-Endpunktkanal-Zahl und eines Ein-Bit-Endpunktkanal-Steuerwortzeigers indiziert. Wie dargestellt ist, zeigt eine Endpunkt-Zahl von null, EP0 932 auf das erste Paar Steuerwörter in der Endpunktkanal-Anordnung 922. In gleicher Weise zeigt EP1 auf das zweite Paar Steuerwörter, und EP2 zeigt auf das dritte Paar usw. EP15 zeigt auf das letzte Paar der Steuerwörter in der bevorzugten Ausführungsform.

Die Steuerwörter umfassen die gesamte verlangte Information, um einen aktiven Endpunktkanal zu betreiben. Wie in dem Block 924 dargestellt ist, weist jedes Steuerwort die Information über den Zustand, den Parameter, die Adresse und den Umfang eines Endpunktkanals auf. Es existieren zwei solche Einträge für jeden Endpunktkanal. Mittels des Wertes des Endpunktkanal-Steuerwortzeigers CWP wird das Steuerwort bestimmt, auf welches als nächstes zugegriffen wird. Unmittelbar nach dem Empfang eines IN-, OUT- oder SETUP-Tokens an einem gegenwärtig in einem Ready-Zustand befindlichen Endpunktkanal wird das Steuerwort des ausgewählten Endpunktkanals am gegenwärtigen Steuerwort-Zeiger gelesen, um zu bestimmen, wie auf den Token zu antworten ist. Wenn am Ende einer Übertragung der Abschluß-Status geschrieben ist, wird das Steuerwort mit dem gegenwärtigen Status aktualisiert, das CWP wird inkrementiert, und das andere Steuerwort wird gelesen. Wenn ein Zwischenstatus geschrieben wird, wird nur das signifikanteste Byte des Steuerworts aktualisiert. Das Format des Steuerwortes ist im folgenden dargestellt:

Bits	Funktionsbezeichnung
31-28	Buffer State (Pufferzustand)
27	(Reserviert)
26-16	Buffer Address BA (Pufferadresse)
15	Buffer Page BP (Pufferseite)
14	Buffer Rounding (Pufferrunden)
13	Toggle Enable TE (Umschaltaktivierung)
12	Type Isochronous TI
11	Interrupt on Packet Complete (Interrupt auf Paketabschluß)
10	Interrupt on Packet Error IE (Interrupt auf Paketfehler)
9 0	Byte Count BC (Byte-Zähler)

Der 4-Bit-Pufferzustand BS ist ein Paket des Steuerwortes, welches für die Anwendung der Erfindung relevant ist. Der Pufferzustand wird mittels Software und mittels des Endpunkt-Controllers (EPC) aktualisiert, um die Steuerung dieses Puffers und zwischen den Puffern zu synchronisieren. Der EPC schreibt den die Nutzung des Puffers anzeigenden Status. Dieses Feld wird mittels Firmware für den Knoten auf Output Ready, Input Ready, Setup Ready, Skip, Stall und Disable (siehe unten) gesetzt. Alle anderen Werte werden mittels des EPC gesetzt. Der STALL- und der Disable-Wert können von dem EPC bei Fehlerbedingungen geschrieben werden. Der Puffer kann mittels Knoten-Firmware wieder beansprucht werden, wenn der Pufferzustand irgendein Complete (Fertigstellungs-) – Wert ist. Dieses umfaßt den Eingangs-/Ausgangs-Complete-Status und die Fehlercode.

Der Pufferzustand BS kann die folgenden Werte annehmen:

0: Skip. Dieser Wert wird zusammen mit dem Output Ready-Befehl genutzt. Skip führte dazu, daß das nächste empfangene Datenpaket ignoriert wird, und daß der Steuerwort-Zeiger inkrementiert wird. Für Steuer-Endpunkte führt es weiterhin dazu, daß der Endpunktkanal-Zustand von dem Input/Ready-Zustand in den Output/Input-Idle (Leerlauf)-Zustand geschaltet wird. Das Setzen von TE und TI bestimmt, welcher Handshake zurückgesendet wird, und bestimmt den nächsten Zustand des Endpunktkanals. Ein NAK wird zurückgesendet, wenn entweder TE oder TI gesetzt ist. Wenn TI und nicht TE gesetzt ist, wird kein Handshake zurückgesendet. Wenn TE gesetzt ist, wird ein ACK zurückgesendet, und der komplette Status wird geschrieben, wenn ein Paket mit einem geeigneten PID empfangen wurde.

1: Input Ready IRDY. Dieser Wert zeigt an, daß in den Puffer Daten geschrieben wurden, um diese als Reaktion auf einen IN-Token zu übertragen.

2: Output Ready ORDY. Dieser Wert zeigt an, daß der Puffer für das Schreiben zugeordnet wurde.

3: Setup Ready SRDY. Dieser Wert zeigt an, daß der Puffer für das Schreiben zugeordnet wurde. Dieses wird an einem Steuerendpunkt nur während der ersten Übertragung von Steuer-Lese- und Steuer-Schreib-Operationen genutzt. Schreib nur in das Steuerwort 0.

4: Output Wait OWT. Dieser Wert zeigt an, daß ein Fehler während des Empfangs der OUT-Daten von dem Host aufgetreten ist. Dieses umfaßt das Erkennen eines Bit-Überlauf-Fehlers. Da dies als ein Zwischenstatus betrachtet wird, wird ein erneuter Versuch erwartet. Ausnahme ist der Fall eines isochronen OUT-Endpunktes, bei dem es als ein Fertigstellungs-Status behandelt wird.

5: CRC Error CRCE. Dieser Wert zeigt an, daß am Ende einer Out-Transaktion ein schlechtes CRC erkannt wurde. Da dies als ein Zwischenstatus betrachtet wird, wird ein erneuter Versuch erwartet. Die Ausnahme ist der Fall eines isochronen Out-Endpunktes, bei dem es als ein Fertigstellungs-Status behandelt wird.

6: Buffer Error BUFE. Dieser Wert zeigt an, daß während einer OUT-Transaktion, Daten von dem USB schneller empfangen wurden, als sie in den Speicher geschrieben werden konnten. Während einer IN-Transaktion wurden die Daten aus dem Speicher nicht schnell genug gelesen, um mit der USB-Datenrate mitzuhalten. An diesem Endpunkt wird eine Wiederholung erwartet.

7: Input Wait IWT. Dieser Wert zeigt an, daß der Puffer durch einen Endpunkt-Controller gehalten wird, und daß der Puffer auf die nächste Gelegenheit wartet, um zu versuchen oder erneut zu versuchen, diesen Puffer zu übertragen. Dieses tritt auf, wenn eine Quittierung fehlt.

8: Input Complete ICMP. Dieser Wert zeigt den erfolgreichen Abschluß dieser Eingabeoperation an. Der Puffer kann freigegeben werden.

9: Output Complete OCOMP. Dieser Wert zeigt an, daß der Puffer mit Daten voll ist, und daß kein Fehler erkannt wurde.

10: Token Error TKNE. Dieser Wert zeigt an, daß ein Token des falschen Typs für den momentanen Kontext dieses Steuer-Endpunktkanals empfangen wurde. Wenn dieses ein Ausgabepuffer war, wird der Endpunkt gehalten. Wenn dies ein Eingabepuffer war, wird ein NAK gesendet, und der Zustand des anderen Steuerwortes wird untersucht. Dieses wird als Abschlußstatus betrachtet.

11: Isochronous Synchronization Error SYNE. Dieser Wert zeigt an, daß Daten in einem Puffer mit TE = 1 und TI = 1 empfangen wurden, wobei der CWP nicht gleich dem LSB des FNR war. Dies wird als ein Abschlußstatus betrachtet.

12: Data Error DATE. Dies tritt für OUT-Daten auf, wenn CW.BR nicht gesetzt ist, und der vom Host empfangene Datenumfang geringer als erwartet war, oder der zugeordnete Puffer nicht groß genug war, um die OUT-Daten zu speichern. Für IN-Daten wird dieses gesetzt, wenn ein Pufferfehler auftritt, und CW.TE nicht gesetzt oder CW.TI gesetzt war. Dies wird als ein Abschlußstatus betrachtet.

13: (Reserviert).

14: Halt. Dieses bringt den gegenwärtigen Endpunktkanal dazu, seinen HALT-Zustand einzunehmen, oder zeigt an, daß der Endpunkt den HALT-Zustand eingenommen hat.

15: Disable. Dieses bringt den momentanen Endpunktkanal dazu, seinen DISABLE-Zustand einzunehmen, oder zeigt an, daß der Endpunktkanal den DISABLE-Zustand eingenommen hat.

Die Puffer-Adresse BA hält die unteren 11 Bits der Pufferadresse. Puffer müssen an einer ausgerichteten Doppelwort-Grenze beginnen und können an jeder Byte-Grenze enden. Am Ende aller abgeschlossenen Übertragungen wird der verbleibende Wert der Pufferadresse geschrieben. Dieses Feld wird am Ende von nicht vollständig ausgeführten Übertragungen nicht überschrieben, z. B. wenn der Status auf Eingangs- oder Ausgangswarten geschrieben ist. Für OUT-Puffer ist das erste gespeicherte Byte das dem PID folgende Byte. Ähnlich ist es für IN-Puffer. Die Adresse zeigt auf das erste Byte der zu übertragenden Daten, d. h. das Byte, welches im Paket unmittelbar auf das PID folgt. Der CRC wird automatisch berechnet und an das Paket angehängen, es sei denn, ein IN-Token wird vom IS\_RDY-Zustand empfangen, und in dem Steuerwort ist das IS-Feld nicht auf 1 gesetzt.

Das Puffer-Seiten-Bit, BP, wählt zwischen einem Puffer-Seiten-A-Register, BPA 912, und einem Puffer-Seiten-B-Register, BPB 916 aus. Das BP wählt das BPA in einem Null-Zustand und wählt das BPB in einem Zustand, in dem es als die obere Speicheradresse des gegenwärtigen Eingangs-/Ausgangspuffers benutzt wird.

Das Pufferrundungsbit BR wird für Out-Transaktionen genutzt. Es zeigt an, ob ein empfangenes Datenpaket kleiner



als der Datenpuffer sein kann. Wenn das BR null ist, sollte das Datenpaket den definierten Datenpuffer exakt füllen. Ein Pufferüberlauf- oder Pufferunterschreitungs-Status wird angezeigt, wenn das Paket nicht exakt in den Puffer paßt.

Wenn das BR zusammen mit einem Ausgangsabschluß- oder Datenfehler-Pufferstatus gelesen wird, zeigt das BR den Wert des empfangenen PIDs an. Es ist null für DATA0 und 1 für DATA1.

- 5 Das Toggle-Enable-Bit (Umschaltaktivierungs-Bit), TE, wird benutzt, um den Steuerwort-Zeiger für ein OUT-Token mit dem Datenpaket-Identifizierer, PID, welcher einen Wert von entweder DATA0 oder DATA1 hat, zu vergleichen, und wird benutzt, um einen Datenpaket-Identifizierer für einen IN-Token zu erzeugen.

- 10 Wenn das TE in einem Null-Zustand ist, wird das Daten-Toggle-PID einer OUT-Transaktion nicht überprüft. Das Daten-Toggle-PID wird auf der Basis des gegenwärtigen Wertes des CWP als Antwort auf IN-Token erzeugt. Das CWP wird nach jeder Transaktion inkrementiert (invertiert), auch wenn ein ACK nicht empfangen wurde. Für Isochrone Endpunkte werden OUT-Daten nicht in dem Speicher geschrieben, bis das CWP gleich dem letzten signifikanten Bit der Frame-Zahl, FNR, ist. Dieses hat keinen Einfluß auf die als Antwort auf die IN-Token gesendeten Daten. Das Datenumschalten sollte nicht für Puffer aktiviert werden, die zugeordnet wurden, um Daten mit einem SETUP-Token zu empfangen.

- 15 Wenn das TE in einem Eins-Zustand ist, wird das Daten-Toggle-PID einer OUT-Transaktion geprüft. Das Daten-Toggle-PID wird auf der Basis des gegenwärtigen Wertes des CWP als Antwort auf IN-Token erzeugt. Für OUT-Transaktionen verursacht das TE, daß ein ACK als Antwort auf die OUT-Pakete gesendet wird, dessen PID-Wert (DATA0 oder DATA1) nicht mit dem CWP übereinstimmt. Für IN-Transaktionen wird das CWP nur nach jeder Transaktion inkrementiert (invertiert), wenn innerhalb der Antwortzeitperiode ein ACK-Handshake zurückgesendet wurde. Für Isochrone Endpunkte werden die OUT-Daten immer in den Speicher geschrieben. Wenn jedoch das letzte signifikante Bit der FNR nicht gleich dem CWP ist, wird der ISO-Sync-Fehlerpufferzustand zurückgeschrieben.

Das Isochron-Typenbit TI zeigt an, ob der Endpunktkanal isochron ist. Wenn das TI in einem Eins-Zustand ist, sind die Daten isochron. Wenn das TI in einem Null-Zustand ist, kann der Endpunktkanal ein Steuer-, Bulk- oder Interrupt-Endpunktkanal sein.

- 25 Ein Interrupt-Paketabschluß-Bit, IC, wird immer gesetzt, wenn der Pufferstatus nach einem Paketfehler auf Eingangs- oder Ausgangsabschluß geschrieben wird, oder wenn der Endpunktkanal verstopft ist.

Ein Interrupt-Paketfehler-Bit, IE, wird immer gesetzt wenn der Pufferstatus auf einen der Puffer-Fehlercodes geschrieben wird, oder wenn der Endpunkt verstopft ist.

- 30 Wenn das IC und der IE gesetzt sind, veranlassen die CRC- und Bit-Stuff-Fehler, daß die entsprechenden Puffer nicht in einer normalen Operation erneut genutzt werden. Dieses ermöglicht die nachfolgende diagnostische Überprüfung der entsprechenden Puffer.

- 35 Byte-Zählbits, BC, zeigen den Umfang des zu nutzenden Datenpuffers an. Ein Puffer muß an einer geraden Doppelwortgrenze beginnen, kann jedoch an jeder beliebigen Byte-Grenze enden. Für IN-Transaktionen zeigt das BC den Umfang der zu übertragenden Daten an. Für OUT-Transaktionen zeigt das BC den Umfang des Puffers an, in welchem die Daten empfangen werden können. Das BC kann eine Null halten, um ein Null-Längen-Datenpaket anzuzeigen. Am Ende aller abgeschlossenen Übertragungen wird der Restwert des Byte-Zählers geschrieben. Am Ende einer nicht abgeschlossenen Übertragung, z. B., wenn der Status Eingangs- oder Ausgangswarten anzeigt, wird dieses Feld nicht überschrieben.

- 40 Bei einer bevorzugten Ausführungsform weist das Endpunkt-Befehl-Register, ECR 918, die Breite eines Wortes auf, und ist ein Lese-/Schreibregister. Das ECR 918 wird genutzt, um Befehle an einem spezifizierten Endpunkt auszuführen. Durch ein Reset wird das Register geleert. Das ECR-Registerformat ist im folgenden dargestellt:

Bits	Funktionsbezeichnung
15-8	(Reserviert)
45 7-4	Endpoint Pipe Number (EPN) (Endpunktkanal-Zahl)
3-0	Endpoint Command (CMD) (Endpunkt-Befehl)

- 50 Die Endpunktkanal-Zahl EPN bestimmt den Endpunktkanal, auf welchem der Befehl ausgeführt wird. Bei einer bevorzugten Ausführungsform kann dieses Register in die Hochgeschwindigkeits-Bus-Schnittstellenlogik implementiert sein. Die Daten laufen über die asynchrone Schnittstelle. Zwei Signale werden vom Endpunkt-Controller zurückgeschickt, wobei eines anzeigt, daß der Befehl ausgeführt wurde, und daß der Befehlswert auf "0" gesetzt werden sollte, und wobei ein weiteres anzeigt, daß der Befehl nicht ausgeführt werden konnte, und daß es auf "F" gesetzt werden sollte.

- 55 Der Endpunkt-Befehl CMD kann die folgenden Werte annehmen:

0000: Complete. Dieser Wert zeigt an, daß der vorübergehende Befehl abgeschlossen ist, und daß der Endpunkt-Controller zur Aufnahme eines anderen Befehls bereit ist.

- 60 0001: Input Enable (CMD.IENB). Dieser Wert veranlaßt den spezifizierten Endpunktkanal aus dem Stall (Halt)- oder Disable-Zustand in den Input Idle-Zustand I\_IDL zu gehen. Der Steuerwort-Zeiger für den spezifizierten Endpunktkanal wird auf 0 zurückgesetzt. Wenn der momentane Zustand nicht der Disable- oder Stall-Zustand ist, wird ein "F" zurückgeschickt, um anzuzeigen, daß versucht wurde, einen nicht erlaubten Befehl an diesem Endpunkt auszuführen, und daß der Befehl ignoriert wird.

- 65 0010: Input Ready (CMD.IRDY). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den Input Ready-Zustand I\_RDY zu gehen, wenn dieser momentan in dem Input Idle-Zustand I\_IDL ist. Dieser Befehl wird ausgegeben, nachdem ein Steuerwort mit einem Input Ready-Pufferzustand I\_RDY in die Endpunkt-Anordnung des entsprechenden Endpunktes geschrieben wurde. Dieser Befehl veranlaßt nicht, daß das Steuerwort im Speicher gelesen wird; es erlaubt hingegen, daß es als Antwort auf den Empfang eines IN-Tokens gelesen wird. Wenn der gegenwärtige Zustand des End-



punktes nicht Input Idle I\_DL ist, wird der Befehl ignoriert. Wenn der gegenwärtige Zustand der Output-, der Setup-, der Disable- oder der Stall-Zustand ist, wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, einen nicht ausführbaren Befehl an diesem Endpunkt auszuführen.

0011: Input Isochronous Ready (CMD.ISRDY). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den Input Isochronous Ready-Zustand IS\_RDY zu gehen, wenn er momentan in dem Input Idle-Zustand I\_IDL ist. Dieser Befehl wird ausgegeben, nachdem ein Steuerwort mit dem Input Ready-Puffer-Zustand in die Endpunkt-Anordnung für den entsprechenden Endpunkt geschrieben wurde. Dieser Befehl veranlaßt nicht, daß das Steuerwort in dem Speicher gelesen wird; es erlaubt hingegen, daß es als Antwort auf den Empfang eines IN-Tokens gelesen wird. Es wird erwartet, daß das Typenfeld des Steuerwortes (TI) auf Isochron gesetzt ist. Dieses wird jedoch nicht überprüft. Wenn der gegenwärtige Zustand des Endpunktes nicht Input Idle I\_IDL ist, wird der Befehl ignoriert. Wenn der gegenwärtige Zustand der Output-, Setup-, Disable- oder der Stall-Zustand ist, wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, einen nicht ausführbaren Befehl an diesem Endpunkt auszuführen.

0100: Output Enable (CMD.OENB). Dieser Wert veranlaßt den spezifizierten Endpunktkanal aus dem Stall- oder Disable-Zustand in den Output Idle-Zustand O\_IDL zu gehen. Der Steuerwort-Zeiger für den spezifizierten Endpunkt wird auf 0 gesetzt. Wenn der gegenwärtige Zustand nicht der Disable- oder Stall-Zustand ist, wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, an diesem Endpunkt einen nicht ausführbaren Befehl auszuführen, und der Befehl wird ignoriert.

0101: Output Ready (CMD.ORDY). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den Output Ready-Zustand O\_RDY zu gehen, wenn er gegenwärtig in dem Output Idle-Zustand O\_IDL ist. Dieser Befehl wird ausgegeben, nachdem ein Steuerwort mit einem Output Ready-Zustand in die Endpunkt-Anordnung für den entsprechenden Endpunkt geschrieben wurde. Dieser Befehl verursacht nicht, daß das Steuerwort in dem Speicher gelesen wird; es erlaubt hingegen, daß es als Antwort auf den Empfang eines OUT-Tokens gelesen wird. Wenn der gegenwärtige Zustand des Endpunktes nicht Output Idle ist, wird der Befehl ignoriert. Wenn der gegenwärtige Zustand der Input-, der Stall- oder der Disable-Zustand ist, dann wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, einen nicht ausführbaren Befehl an diesen Endpunkt auszuführen.

0110: Output Isochronous Enable (CMD.OSENB). Dieser Wert veranlaßt den spezifizierten Endpunktkanal aus dem Stall- oder Disable-Zustand in den Output Isochronous Idle-Zustand OS\_IDL zu gehen. Der Steuerwort-Zeiger führt den spezifizierten Endpunkt auf 0 gesetzt. Wenn der gegenwärtige Zustand nicht der Disable- oder Stall-Zustand ist, wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, einen nicht ausführbaren Befehl an diesem Endpunkt auszuführen, und der Befehl wird ignoriert.

0111: Output Isochronous Ready (CMD.OSRDY). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den Output Isochronous Ready-Zustand OS\_RDY zu gehen, wenn dessen momentaner Zustand der Output Isochronous Idle-Zustand OS\_IDL ist. Dieser Befehl wird ausgegeben, nachdem ein Steuerwort mit einem Output Ready-Zustand in die Endpunkt-Anordnung für den entsprechenden Endpunkt geschrieben wurde. Dieses Kommando veranlaßt nicht, daß das Steuerwort in dem Speicher gelesen wird; es erlaubt hingegen, daß es als Antwort auf den Empfang eines OUT-Tokens gelesen wird. Wenn der momentane Zustand des Endpunktes nicht der Output Isochronous Idle-Zustand ist, wird der Befehl ignoriert. Wenn der gegenwärtige Zustand der Input-, Stall- oder Disable-Zustand ist, wird ein "F" zurückgesendet, um anzuzeigen, daß versucht wurde, einen nicht ausführbaren Befehl an diesem Endpunkt auszuführen.

1000: Setup Ready (CMD.SRDY). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den Setup Ready-Zustand S\_RDY zu gehen. Dieser Befehl wird ausgegeben, nachdem ein Steuerwort mit einem Output Ready-Zustand in die Endpunkt-Anordnung für den entsprechenden Endpunkt geschrieben wurde. Dieser Befehl veranlaßt nicht, daß das Steuerwort in dem Speicher gelesen wird; es erlaubt hingegen, daß es als Antwort auf den Empfang eines SETUP-Tokens gelesen wird. Wenn der gegenwärtige Zustand des Endpunktes nicht der Stall- oder Disable-Zustand ist, wird der Befehl ignoriert.

1001: Disable (CMD.DISAB). Dieser Wert veranlaßt den spezifizierten Endpunktkanal in den DISABLE-Zustand zu gehen. Dieses erfolgt unmittelbar aus jedem Zustand. Wenn der Endpunktkanal gegenwärtig aktiv ist, wird der Status in die Steuerung geschrieben, so daß angezeigt ist, daß der Endpunkt nicht aktiviert ist.

1010: Stall (CMD.HALT). Dieser Wert veranlaßt den spezifizierten Endpunkt aus einem beliebigen Zustand in den Stall-Zustand HALT zu gehen. Dieses erfolgt unmittelbar aus jedem Zustand. Wenn der Endpunktkanal gegenwärtig aktiv ist, wird der Status in die Steuerung geschrieben, so daß angezeigt ist, daß der Endpunkt blockiert ist.

1011–1101: Reserviert. Diese Werte verursachen nicht die Ausführung von Aktionen und senden eine 0 zurück.

4110: Toggle CWP. Dieser Wert veranlaßt ein Umschalten des Steuerwort-Zeigers. Dieses wird im Zusammenhang mit Steuerübertragungen mit einer ungeraden Anzahl gesendeter oder empfangener Pakete während einer Datenphase genutzt.

1111: Error. Dieser Wert zeigt an, daß versucht wurde, einen für den Zustand des gegenwärtigen Endpunktkanals nicht ausführbaren Befehl auszuführen, und daß ein neuer Befehl aufgenommen werden kann.

Der für die Erfindung in der vorliegenden Anwendung relevante Teil des UDR ist der 8-Bit-Endpunktkanal-Zustand. Dieser Zustand wird für jeden Endpunkt gespeichert, der für den gegenwärtig indizierten Endpunktkanal zurückgesendet wird. Wenn ein ungültiger Endpunktkanal indiziert ist, wird eine 0 zurückgesendet. Das UDR-Registerformat ist im folgenden dargestellt:

Bits	Funktionsbezeichnung
7	Control (CTL)
6	Control Word Pointer (CWP) (Steuerwort-Zeiger)
5–4	(Reserviert)
3–0	Endpoint Pipe State (EPS) (Endpunktkanal-Zustand)

Control CTL: Dieses wird geltend gemacht, wenn dieser Endpunktkanal in der Lage ist, Steuersequenzen aufzunehmen, die mit SETUP-Token zu tun haben. Dieses Bit wird gesetzt, nachdem der CMD.SRDY für diesen Endpunktkanal ausgeführt wurde, und verbleibt gesetzt, bis der Endpunkt deaktiviert (Disable) wird.

Control Word Pointer CWP: Wenn das Steuerwort-Bit 0 ist, wird auf das Steuerwort 0 dieses Endpunktes in der Endpunkt-Anordnung als nächstes zugegriffen. Wenn das Steuerwort 1 ist, wird auf das Steuerwort 1 für diesen Endpunkt als nächstes zugegriffen. Dieses ist auf 0 gesetzt, wenn der Endpunkt aktiviert ist.

Der Endpunktkanal-Zustand (EPS) kann die folgenden Werte annehmen:

Wert	Zustand
0	Disabled DIS
1	Input Idle I_IDL
2	Input Ready I_RDY
3	Input Active I_ACT
4	Stalled HALT
5	Input Missed ACK I_MAK
6	Input Isochronous Ready IS_RDY
7	Input Isochronous Active IS_ACT
8	Setup Ready S_RDY
9	Output Idle O_IDL
10	Output Ready O_RDY
11	Output Active O_ACT
12	Setup Active S_ACT
13	Output Isochronous Idle OS_IDL
14	Output Isochronous Ready OS_RDY
15	Output Isochronous Active OS_ACT

Bei einer bevorzugten Ausführungsform ist zu jeder Zeit nur ein Endpunktkanal aktiv.

Bei einer bevorzugten Ausführungsform bewahrt jeder Endpunktkanal einen Zustand mit definierten Bedingungen für die Übertragung zwischen den Zuständen, welche eine implizierte Hardware/Software-Synchronisation umfassen. Übergänge zwischen Endpunkt-Zuständen erfolgen:

- auf einen expliziten Software-Befehl an einen Endpunkt durch das Endpunkt-Befehlsregister ECR;
- durch das Lesen des Pufferstatus des anderen Puffers in dem anderen Steuerwort für diesen Endpunkt am Ende einer Übertragung; und
- als Ergebnis von Ereignissen auf dem USB, wie der Empfang von IN, OUT- und SETUP-Token und von ACK-Handshakes.

Fig. 10 zeigt die vereinfachte Endpunktkanal-Zustandsmaschine für die Handhabung von SETUP-Transaktionen.

Der gegenwärtige Endpunktkanal-Zustand ist in einem Endpunkt-Zustandsregister sichtbar, auf welches mit einem USB-Index- und einem -Daten-Register zugegriffen werden kann. Bei einer bevorzugten Ausführungsform ist zu jeder Zeit stets höchstens ein Endpunktkanal in irgendeinem der aktiven Zustände. Dies ermöglicht das gemeinsame Nutzen der aktiven Zustandslogik durch alle Endpunktkanäle.

Wie in Fig. 10 gezeigt ist, kann ein Endpunktkanal die folgenden Zustände aufweisen:

Disable. Der DIS-Zustand 1010 wird auf ein Reset, auf einen Disable-Befehl an diesen Endpunktkanal, und auf das Erkennen eines Disable-Zustandes in dem Pufferzustand des Steuerwortes eingenommen. Während dieser Endpunktkanal in diesem Zustand ist, wird jeder an diesem Endpunkt empfangene Token ohne Handshake zurückgesendet. Eine Software-Intervention ist notwendig, um den Endpunktkanal aus diesem Zustand zu entfernen. Typischerweise geschieht dies durch einen Enable-Befehl (Aktivierungsbefehl) an einen der Idle-Zustände.

Halt/Stall: Der HALT/STL-Zustand 1070 wird auf den Stall-Befehl an diesen Endpunkt, auf das Erkennen eines Stall-Zustandes in dem Pufferzustand eines Steuerwortes, und bei bestimmten Fehlerbedingungen eingenommen. Während dieser Endpunktkanal in diesem Zustand ist, wird jeder an diesem Endpunkt empfangene Token als ein Stall-Handshake zurückgesendet, um anzuzeigen, daß der Endpunkt gegenwärtig blockiert ist. Eine Software-Intervention ist notwendig, um den Endpunktkanal aus diesem Zustand zu bringen. Typischerweise geschieht dies durch einen Enable-Befehl an einen der Idle-Zustände.

Setup Ready S\_RDY. Dieser Zustand 1020 wird eingenommen, nachdem dieser Endpunktkanal einen Puffer erhalten hat, in welchen die Daten zu schreiben sind. Dieser Zustand wird auf das Erkennen des Setup Ready-Pufferzustandes oder mittels eines Setup Ready-Befehls eingenommen, nachdem das geeignete Wort in die Endpunkt-Anordnung geschrieben wurde.

Setup Active S\_ACT. Dieser Zustand 1030 wird auf den Empfang eines Setup-Tokens aus dem Setup Ready-Zustand eingenommen. In diesem Zustand werden die Daten vom USB in den Speicher geschrieben. Am Ende einer Ausgangsübertragung wird der Status in das Steuerwort am CWP geschrieben, das CWP wird invertiert, und das andere Steuerwort für diesen Endpunktkanal wird gelesen. Es wird der passende Übergang in den Output Idle-, Output Ready, Input Idle oder Input Ready-Zustand gemacht. Er hängt von der Art der Steuerübertragung und dem gelesenen Pufferzustand ab, und basiert weiterhin auf dem Pufferstatus.

Input Idle I\_IDL. In diesem Zustand 1050 erwartet der Endpunkt, daß bald ein Puffer zugeordnet wird. Ein NAK-Handshake wird als Antwort auf einen IN-Token, welcher diesem Endpunktkanal zugeordnet ist, zurückgesendet. Für OUT-Token wird von diesem Endpunktkanal keine Antwort zurückgesendet.

**Input Ready I\_RDY.** Dieser Zustand **1052** wird über einen Input Ready-Befehl an diesen Endpunktkanal oder auf das Erkennen eines anderen Eingangspuffers am Ende einer Übertragung eingenommen. Der Befehl sollte nur gegeben werden, wenn diesem Endpunkt mit Hilfe des Schreibens des Steuerwortes an einem geeigneten Ort in der Endpunkt-Anordnung ein Puffer zugeordnet worden ist. Dieses Steuerwort sollte auf einen Datenpuffer mit Daten zeigen, die an den Host als Antwort auf den nächsten an diesem Endpunktkanal empfangenen IN-Token zurückgesendet werden sollen. Der PID-Wert wird als DATA0 übertragen, wenn der CWP 0 ist und wird als DATA1 übertragen, wenn der CWP gleich 1 ist.

**Input Active I\_ACT.** Dieser Zustand **1054** wird auf den Empfang eines IN-Tokens eingenommen, wenn der ausgewählte Endpunkt-Zustand gegenwärtig Input Ready ist. In diesem Zustand werden die Daten von dem Speicher auf den USB übertragen. Auf den Abschluß der Datenübertragung sollte ein ACK-Handshake innerhalb der 16 Bit-Zeit empfangen werden. Am Ende dieses Zeitlimits wird der Puffer-Wartestatus in das Steuerwort zurückgeschrieben, wenn ein ACK nicht empfangen wurde. Die Adress- und Byte-Zähler werden jedoch nicht überschrieben, da eine erneute Übertragung erfolgen wird und die gegenwärtigen Werte erneut gebraucht werden. Wenn ein ACK empfangen wird, wird der Pufferstatus zusammen mit den gegenwärtigen Werten des Adress- und-Byte-Zählers geschrieben. Danach wird das andere Steuerwort an diesem Endpunkt gelesen, um den nächsten Zustand des Endpunktes zu bestimmen, welcher benutzt werden wird, wenn das nächste Mal ein Token an diesen Endpunkt gerichtet ist. Wenn dieser Endpunkt für eine Steuerübertragung genutzt wird, kann eine Übertragung auf den Ausgangs-Ready- oder Output-Idle-Zustand erfolgen, um die Statusphase einer Steuer-Lese-Übertragung zu vervollständigen.

**Output Idle 0\_IDL.** In diesem Zustand **1040** erwartet der Endpunktkanal, daß bald ein Puffer zugeordnet wird. Ein NAK-Handshake wird als Antwort auf ein OUT-Token, welcher diesem Endpunktkanal zugeordnet ist, zurückgesendet. Für IN-Token auf diesem Endpunktkanal wird keine Antwort zurückgesendet.

**Output Ready 0\_RDY.** Dieser Zustand wird eingenommen, nachdem dieser Endpunktkanal einen Puffer erhalten hat, in welchen die Daten geschrieben werden. Dieser kann auf das Erkennen eines anderen Ausgangspuffers am Ende einer Übertragung oder über einen Output Ready-Befehl von dem 0\_IDL-Zustand **1040** eingenommen werden, nachdem das passende Wort in die Endpunkt-Anordnung geschrieben wurde.

**Output Active 0\_ACT.** Dieser Zustand **1044** wird auf den Empfang eines OUT-Tokens eingenommen. In diesem Zustand werden die Daten aus dem USB in den Speicher geschrieben. Am Ende einer Ausgangsübertragung wird der Status auf das Steuerwort am CWP geschrieben. Wenn die Übertragung nicht erfolgreich war, d. h. ein CRC- oder Bit-Blockier-Fehler wurde erkannt, wird dieses als der Pufferstatus geschrieben. Wenn die Übertragung erfolgreich war, wird der Output Complete-Pufferstatus geschrieben, wird der CWP invertiert, und wird das andere Steuerwort für diesen Endpunktkanal gelesen. Die geeignete Übertragung wird basierend auf den Pufferzustand ausgeführt. Wenn dieser Endpunkt für eine Steuerübertragung genutzt wird, kann ein Übergang in den Input Ready- oder Input Idle-Zustand erfolgen, um die Statusphase eines Steuer-Schreib-Überganges zu vollenden.

Nach einem Reset sind alle Endpunkte inaktiviert. Ein Endpunkt ist mittels eines Aktivierungsbefehls an den Endpunkt aktiviert. Dieses sollte nur dann getan werden, wenn die Eingangs- und/oder Ausgangspuffer diesem Endpunkt bald zugeordnet werden. Alle Endpunkte, die gegenwärtig nicht einem Endpunktkanal zugeordnet sind, werden als inaktiviert betrachtet.

Die Software ordnet den Endpunktkanälen Puffer nach Bedarf zu. Nachdem jeder Puffer zugeordnet ist, führt sie den Input Ready- oder Output Ready-Befehl an dem Endpunktkanal aus. Wenn der Endpunktkanal gegenwärtig im Idle-Zustand ist, nimmt er den geeigneten Ready-Zustand ein. Wenn der Endpunktkanal gegenwärtig im Ready- oder Active-Zustand ist, werden alle Befehle ignoriert, mit Ausnahme der Disable- und Stall-Befehle an den Endpunktkanal, welche dieser unverzüglich ausführt. Dieses Schreiben muß von der Software ausgeführt werden, um in der Lage zu sein, potentielle Synchronisationsprobleme zu verhindern, da das Handling des UNIM-Endpunktkanals asynchron zum Betrieb der Software ist.

Weil bis zu zwei Puffer jedem Endpunkt zugeordnet werden können, kann der nächste Puffer vorbereitet werden, während der gegenwärtige Puffer abgearbeitet wird, um die Datenübertragung zu beschleunigen, und um unnötige NAK-Handshakes zu vermeiden.

Ausgehend von der vorhergehenden Beschreibung und aus Sicht einer funktionellen Steuer-Software wird der Betrieb eines Gerätes als ein Steuerendpunkt im Detail beschrieben. Endpunkt 0 ist typischerweise ein Steuerendpunkt. Zusätzlich ermöglicht ein Abbildungsmechanismus, daß andere Endpunkte Steuerendpunkte werden. Beispielsweise sollte der SRDY-Befehl für die nächste Steuerübertragung zusammen mit dem Puffer oder dem Paket der Status-Übertragung der gegenwärtigen Steuerübertragung geschrieben werden. Wenn während einer Datenphase einer Steuer-Lese- oder Steuer-Schreib-Operation eine ungerade Anzahl von Datenpaketen gesendet wird, bevor die Statusphase gesendet ist, muß Software den Steuerwort-Zeiger anpassen, weil die Statusphase in das Steuerwort 1 abgelegt werden muß, und weil der nächste SRDY-Puffer in das Steuerwort 0 abgelegt werden muß. Um dieses zu erreichen, kann die Software einen SKIP-Puffer einfügen oder ein Toggle-CWP-Befehl erzeugen. Die Software kann dann den Ready-Befehl für den Statusphasen-Puffer ausgeben.

Sollte an einem Steuer-Endpunkt, für welchen ein SRDY-Puffer momentan nicht verfügbar ist, ein SETUP-Token erkannt werden, wird ein beträchtlicher Aufwand betrieben, um den SETUP-Token zu empfangen. Der SETUP-Token wird in der Datenleitung eines Speicherkanals gespeichert, während darauf gewartet wird, daß die Software einen Puffer zuordnet.

Wenn ein gültiger SETUP empfangen wird, wird ein Interrupt über ein Interrupt-Bit durchgesetzt. Als Antwort auf den Interrupt sollte die Software einen Puffer in das Steuerwort 0 schreiben, und sollte einen SRDY-Befehl ausgeben. Hierdurch wird das anhängige SETUP in den Speicher geschrieben. Der SRDY-Befehl verursacht, daß der Endpunkt als ein Steuerendpunkt betrachtet wird, und er bleibt gesetzt, bis ein Eingangsaktivierungs- oder Ausgangsaktivierungsbefehl ausgeführt wird. Der SDRY-Puffer sollte so schnell wie möglich bereitgestellt werden, da es den Datenweg für jede weitere Datenübertragung blockiert, wenn der Interrupt durchgesetzt ist. Allen über das USB-Netzwerk empfangenen Token wird während des Wartens auf den Puffer ein NAK zurückgesendet.

Nachdem ein Setup empfangen wurde, muß es überprüft werden, und eine passende Antwort bestimmt werden. Zu

diesem Zeitpunkt stellt die Firmware passend zusätzliche Eingangs- und Ausgangspuffer zur Verfügung. Wenn Setup ein Befehl ist, der nicht durch die Funktion unterstützt wird, wird ein neuer SRDY-Puffer auf das Steuerwort 0 gegeben, und ein SRDY-Befehl wird ausgegeben. Hierdurch wird allen Token an diesem Endpunkt ein NAK-Handshake zurückgesendet, bis das nächste Setup empfangen wird. Während in der Zwischenzeit die Software das Setup analysiert, werden als Antwort auf die Token NAKs gesendet.

Obwohl die Ausführungsformen unter Bezugnahme auf spezielle Gerätestrukturen, Programmierzustände und Programmiermodelle beschrieben wurden, kann die Erfindung auf eine Vielzahl anderer Geräte angewendet werden, welche verschiedene Programmierzustände und -modelle implementieren. Der Fachmann entnimmt den Ausführungsformen, daß viele Modifikationen möglich sind, ohne deren Lehre zu verlassen. All diese Modifikationen sollen von den folgenden Ansprüchen umfaßt sein.

Die Architektur der USB-Knotenschnittstelle ist in dem folgenden Anhang, der Teil der Beschreibung ist, näher spezifiziert.

## Kapitel 1

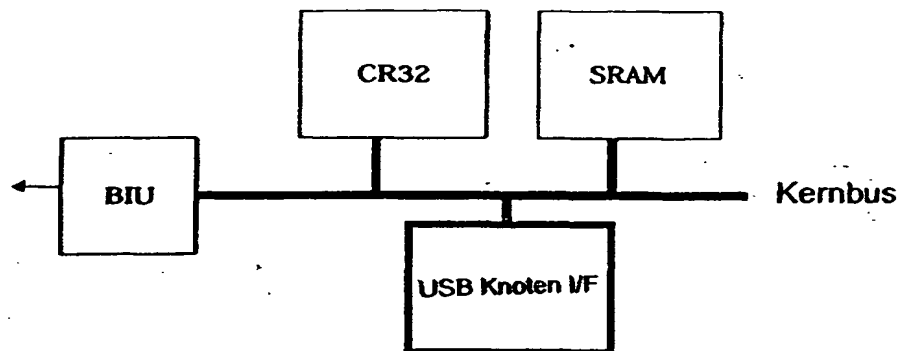
### Einführung

Das USB-Schnittstellenmodul (UNIM) bildet eine Schnittstelle zu dem universellen seriellen Bus für einen Hochgeschwindigkeitsknoten auf einem USB-Bus. Das Modul entspricht der Version 1.0 der USB-Spezifikation.

Das Modul umfaßt einen integrierten USB-Transceiver, eine vollständige serielle Schnittstelleneinrichtung und den Endpunkt-Controller. Der Endpunkt-Controller arbeitet als Master und Slave auf dem Kernbus (Core Bus) und unterstützt alle USB-Endpunktklassen: Steuerung, große Mengen (Bulk), Isochron und Unterbrechung (Interrupt). Der Slavezugang wird dazu verwendet, auf interne Steuerregister zuzugreifen, und der Masterzugang wird dazu verwendet, Endpunktdaten zu und von einem RAM zu übertragen.

Fig. 1-1

#### Anwendungsbeispiel



# DE 199 00 369 A 1

## 1.1 Externe Schnittstelle

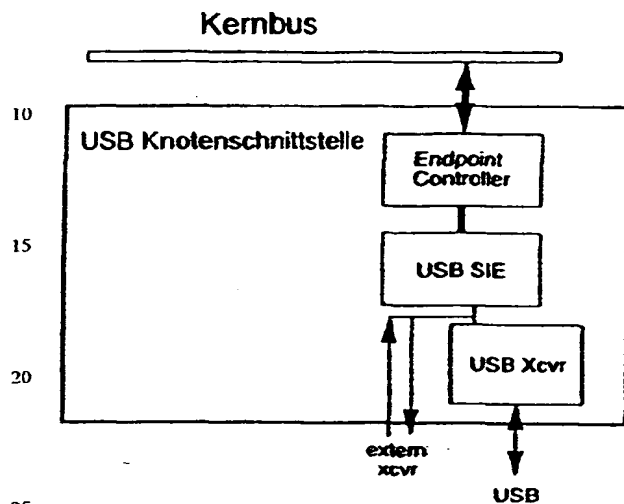
Tabelle 1-1

Modulpins, die mit externen Pins verbunden sind

Symbol	Pin	I/O	Funktion
C48		I	48 MHz Eingangstakt
D+/-		I/O	USB-Daten Plus/Minus – wird verwendet, wenn der interne Transceiver eingesetzt wird. Ein externer 1,5 +/- 5% kOhm Widerstand, der mit einer Spannungsquelle zwischen 3,0 V und 3,6 V bezogen auf die lokale Masse verbunden ist, wird an D+ benötigt, um anzuzeigen, daß es sich um einen Hochgeschwindigkeitsknoten handelt.
USBC		O	1 KHz USB-Bustakt
RC		I	Bezugstakt – Bezugseingangssignal, das mit dem 1 KHz USB-Takt verglichen wird. Das Bezugszählregister wird mit jeder steigenden Flanke dieses Takts erhöht. Der relative Zählwert daraus im Verhältnis zu USB-Rahmennummer kann verwendet werden, um Pakete zu dem USB-Host zu senden, um den 1 KHz USB-Takt schneller oder langsamer zu machen.
SUSPND		O	Die USB-Schnittstelle ist momentan unterbrochen und in einem Niederleistungs-Versorgungsmodus. Verbunden mit einem externen USB-Transceiver, falls vorhanden.
OE		O	Ausgang enable. Aktiv niedrig. Bewirkt, daß der Sender Daten auf den Bus sendet. Verbunden mit einem externen USB-Transceiver, falls vorhanden.
VPO/VMO		O	Differentialsenderdaten an Differentialtransceiver ausgehen. Mit einem externen USB-Transceiver verbunden, falls vorhanden.
RCV		I	Daten empfangen. Ausgang des Differentialempfängers. Mit einem externen USB-Transceiver verbunden, falls vorhanden.
VP/VM		I	Unsymmetrischer Datenempfang – Verwendung zum Erfassen unsymmetrischer Null und Fehlerbedingungen. Verbunden mit einem externen USB-Transceiver, falls vorhanden.
SPEED		I	Geschwindigkeit (SPEED) ist immer hoch (HIGH SPEED) und bei dem momentanen Design auf 1 eingestellt. Verbunden mit einem externen USB-Transceiver, falls vorhanden. Ein Pin für SPEED nur dann zuweisen, wenn eine Option zum Arbeiten sowohl im Niedergeschwindigkeitsmodus als auch im Hochgeschwindigkeitsmodus behalten wird.
CoreBus		I	Verbinden mit Kernbus
USB_INI		I	Interrupt-Signal vom USB-Modus an den Interruptcontroller

Fig. 1-2

Blockdiagramm des USB-Knotenschnittstellenmoduls



## 1.2.1 Endpunktcontroller

Der Endpunktcontroller verarbeitet auf den Endpunkt bezogene Operationen einschließlich des Endpunktkanalzustands, der Pufferung und Übertragungen. Bis zu 16 Endpunktkanäle können zu jeder gegebenen Zeit unterstützt werden. Jeder Endpunktkanal kann in dem Adreßraum des Kernbusses zwei Puffern zugewiesen werden, die für Empfangs- und Sendeoperationen verwendet werden. Wenn einem Endpunktkanal keine Puffer zugewiesen sind, ist der Controller verantwortlich, die richtigen Handshakes (Quittungen) für Eingangs- und Ausgangstoken vorzuschicken.

## 1.2.2 Serielle USB-Schnittstelleneinrichtung

Die serielle USB-Schnittstelleneinrichtung umfaßt USB PHY und USB MAC. USB PHY ist verantwortlich für die NRZI-Codierung, die Erfassung des Anfangs und Endes von Paketen, Bitstuffing und Unstuffing und für die Parallel-Seriell- und Seriell-Parallel-Wandlung. USB MAC ist verantwortlich für das Erkennen und Erzeugen geeigneter Paketformate und für das Verarbeiten der Handshakesequenzen zusammen mit dem Endpunktkanal-Controller. USB MAC ist auch verantwortlich zum Erzeugen und Überprüfen der geeigneten CRCs in den Paketformaten. Es gibt eine eng gekoppelte Schnittstelle zwischen dem USB MAC und dem Endpunkt-Controller.

## 1.2.3 USB-Transceiver

Der USB-Transceiver sieht die analoge Verbindbarkeit zu dem USB-Bus gemäß Kapitel 7 der Universal Serial Bus Specification Version 1.0 vor. Dies umfaßt die notwendige Signalformung und Erfassungsfunktionen. Das Modul unterstützt den Hochgeschwindigkeitsbetrieb. Der Betrieb mit niedriger Geschwindigkeit ist für diese Anwendung kein Erfordernis. Das Modul unterstützt auch einen Niederleistungsbetrieb, bei dem der Transceiver sehr wenig Leistung benötigt (die ganze USB-Einrichtung erhält nur 100 µA) und dennoch der Walk-up-Übergang auf den Bus erfassen kann.

Das Modul unterstützt auch eine Verbindung zu einem externen Stand-Alone USB-Transceiver, wie dem PDIUSBP11 von Phillips.

## 1.2.4 Zukünftige Modulerweiterungen

Da die Architektur des Endpunktcontrollers modular ist, kann die Anzahl der unterstützten Endpunktkanäle verändert werden, indem die Anzahl der gespeicherten Teilendpunktkanal-Zustandsvektoren verändert wird. Zusätzlich kann die Abbildung von

- a) bis zu vier Funktionsadressen
- b) des Tokentyps
- c) der Endpunktnummer

auf die Endpunktkanalnummer über die Kanalabbildungsfunktion programmiert werden.

Ein USB HUB-Schnittstellenmodul kann hinzugefügt werden, um eine HUB-Funktion mit bis zu vier zusätzlichen stromabwärtigen Ports zu ermöglichen. Dieses Modul wird zwischen dem USB SIE-Modul und dem USB-Transceivermodul eingefügt. Dadurch kann die Einrichtung die Verbindungsmöglichkeit zu einem oder mehreren zusätzlichen Peripheriegeräten schaffen.

Die Architektur ermöglicht eine Realisierung eines Stand-Alone-USB-Knotencontrollers mit internem oder externem RAM. Zusätzlich könnte die Architektur für Peripheriegeräte mit unterschiedlichem Leistungsniveau und für einen 16-Bit-Bus oder den Peripheriebus mit einem privaten Speicher verwendet werden. Zusätzlich könnten mehrere solche Zellen in einer Ausführung eingerichtet werden, um zusammengesetzte Multifunktionsgeräte zu unterstützen.

Das Design ist für Niedergeschwindigkeits-Knoten nicht optimiert, es könnte jedoch modifiziert werden, um auch mit Niedergeschwindigkeits-Geräten zu arbeiten.

## Kapitel 2 UNIM Modul

### 2.1 USB-Knotenschnittstelle

Das USB-Knotenschnittstellenmodul (UNIM) bildet die Schnittstelle zu einem universellen seriellen Bus (USB). Das Modul umfaßt einen integrierten USB-Transceiver, eine vollständige serielle Schnittstelleneinrichtung (SIE; Serial Interface Engine) und den Endpunktcontroller. Der Endpunktcontroller arbeitet als Master und Slave auf dem Kernbus und unterstützt alle USB-Endpunktclassen: Steuerung, Bulk, Isochron und Interrupt. Der Slavezugang wird dazu verwendet, auf interne Steuerregister zuzugreifen, und der Masterzugang wird dazu verwendet, Daten zwischen Endpunkten und dem RAM zu übertragen.

#### 2.1.1 Merkmale

- Kompatibilität mit der USB 1.0 Spezifikation
- Unterstützt eine USB-Funktion und bis zu 16 Endpunktkanäle
- Unterstützt alle Klassen der USB-Endpunkte
- Unterstützt Doppelpuffer auf jedem Endpunktkanal
- Unterstützt Betrieb als eine Bus-getriebene Einrichtung
- Optimierte für Hochgeschwindigkeitsknotenbetrieb
- USB-kompatible serielle Schnittstelleneinrichtung (SIE)
- USB-kompatible Hochgeschwindigkeitstransceiver
- Unterstützt Niederleistungs-Aussetzbetrieb
- Unterstützt Aufwecken von Normandy aufgrund USB-Wiederaufnahme
- Unterstützt Auslösung von Wiederaufnahme durch Fern-Weckruf
- Unterstützt Synchronisierung des USB-Takts zu externem Bezugstakt.

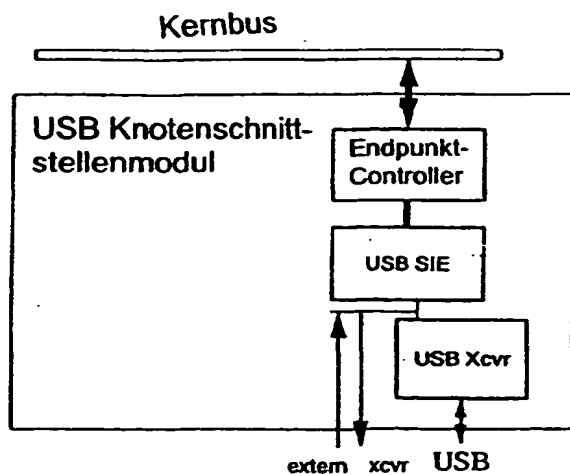
#### 2.1.2 Funktionsbeschreibung

Das USB-Knotenschnittstellenmodul sieht eine Schnittstelle zum USB für verschiedene Anwendungen vor. Der Endpunkt-Controller sieht eine vielseitige Anordnung zum Steuern mehrere USB-Endpunkte vor. Die USB-Knotenschnittstelle umfaßt eine voll anpassungsfähige serielle Schnittstelleneinrichtung (SIE) und einen integrierten Transceiver. Zusätzlich unterstützt sie einen externen Transceiver.

Fig. 2-1 zeigt die obere Ebene dieses Moduls.

Fig. 2-1

Diagramm des USB-Knotenmoduls



#### 2.1.3 Transceiver

Der interne USB-Transceiver umfaßt drei Hauptfunktionsblöcke: Den Differenzialempfänger, den unsymmetrischen



Empfänger und den Sender. Zusätzlich sind eine Bezugsspannung für den unsymmetrischen Empfänger und eine Stromquelle für den Sender vorgesehen.

Normandy unterstützt nur einen Hochgeschwindigkeitstransceiver. Die Leistungserfordernisse dieses Transceivers sind im Kapitel 7 der Universal Serial Bus Specification Version 1.0 angegeben.

- 5 Um die Signalverschiebung zu minimieren, werden Differentialausgangsschwingungen des Senders gut ausgeglichen. Für den Treiber wird eine Anstiegsgeschwindigkeitsschaltung verwendet, um abgestrahltes Rauschen und Übersprechen zu minimieren. Die Treiber unterstützen den TRI-STATE-Betrieb, um einen bidirektionalen Half-Duplex-Betrieb des Transceivers zu ermöglichen.

- 10 Der Differentialempfänger muß im gesamten Gleichtaktbereich arbeiten, und er muß eine Verzögerung aufweisen, die sicher größer als die des unsymmetrischen Empfängers ist, um Spannungsspitzen in der seriellen Schnittstelleneinrichtung (SIE) nach unsymmetrischen Nullen (Single-Ended Zeros) zu vermeiden.

- Unsymmetrische Empfänger sind auf jeder der beiden Datenleitungen vorhanden. Diese sind zusätzlich zu den Differentialempfängern erforderlich, um eine Absolutspannung mit einer Schaltschwelle zwischen 0,8 V und 2,0 V (TTL-Eingänge) zu erfassen. Zur Erhöhung der  $V_{cc}$ -Unterdrückung ohne Spannungsspitzen wird der unsymmetrische Schaltungs-  
15 Bezugswert mit einer Bezugsspannung eingestellt. Ein externer  $1,5 \pm 5\%$  kOhm Widerstand, der mit einer Spannungsquelle zwischen 3,0 V und 3,6 V bezogen auf lokale Masse verbunden ist, ist an D+ erforderlich, um anzuzeigen, daß es sich um einen Hochgeschwindigkeitsknoten handelt.

#### Verwendung eines externen Transceivers

- 20 Es kann auch ein externer Transceiver verwendet werden. Um den Betrieb für einen externen Transceiver zu konfigurieren, muß das externe Transceiver-Enable-Bit (NCR.ETE) gesetzt werden. Wenn es gesetzt ist, gibt es an, daß die Ausgänge ihre "1"-Zustände verlassen sollten, und alle Eingänge sollten gewählt sein.

#### 2.1.4 Serielle Schnittstelleneinrichtung (SIE; serial interface engine)

- Die serielle Schnittstelleneinrichtung des USB besteht aus einer PHY und einer MAC-Ebene. Die PHY-Ebene umfaßt die digitale Taktwiedergewinnungsschaltung, ein digitales Spannungsspitzenfilter, eine Paket-Ende-Erfassungsschaltung und eine bit-stuffing und de-stuffing Logik. Die MAC-Ebene umfaßt die Paketformatierung, CRC-Erzeugung und -Über-  
30 prüfung, Endpunktadressenerfassung, und sie sieht die notwendige Steuerung vor, um NAK, ACK und STALL Antworten zu liefern, die von dem Endpunktcontroller für den spezifizierten Endpunktkanal ermittelt werden. Die SIE ist auch verantwortlich für die Erfassung und Mitteilung von Ereignissen bei Erfassung von USB-spezifischen Ereignissen, wie Reset, Warten und Wiederaufnehmen. Die Ausgangssignale des Senders des UNIM-Moduls, die zu dem Transceiver gehen, sind gut abgeglichen (unter 1 ns), um Verschiebungen auf den USB-Signalen zu minimieren.

#### 3.1.5 Endpunktkanal-Controller (EPC)

- Der Endpunktkanal-Controller (EPC) bildet die Schnittstelle für USB-Funktionsendpunkte vor. Ein Endpunkt ist die letztendliche Quelle oder der Ursprung der Daten. Ein Endpunktkanal sorgt für die Bewegung der Daten zwischen dem  
40 USB und dem Speicher und vervollständigt den Weg zwischen dem USB-Host und dem Funktionsendpunkt. Bis zu 16 solche Endpunktkanäle werden zu jeder gegebenen Zeit unterstützt, jeder mit derselben Funktionsadresse.

- Eine USB-Funktion ist eine USB-Einrichtung, die Informationen auf dem Bus senden und empfangen kann. Eine Funktion kann eine oder mehrere Konfigurationen haben, von denen jede die Schnittstellen definiert, welche die Einrichtung umfaßt. Jede Schnittstelle weist ihrerseits einen oder mehrere Endpunkte auf.

- 45 Jeder Endpunkt ist eine adressierbare Einheit auf dem USB und ist erforderlich, um auf IN- und OUT-Token von dem USB-Host (üblicherweise ein PC) zu reagieren. IN Token geben an, daß der Host angefordert hat, von einem Endpunkt Informationen zu empfangen, und OUT-Token geben an, daß der Host dabei ist, Information zu einem Endpunkt zu senden.

- Bei Erfassung eines IN-Tokens, der an einen Endpunkt gerichtet ist, ist dieser Endpunkt dafür verantwortlich, mit einem Datenpaket zu antworten. Wenn der Endpunkt momentan unterbrochen ist, wird ein STALL-Handshakepaket gesendet. Wenn der Endpunkt enabled ist, jedoch keine Daten vorliegen, wird ein NAK (negative Bestätigung) zum Handshakepaket gesendet.

- Ähnlich ist bei der Erfassung eines OUT-Tokens, der an einen Endpunkt gerichtet ist, der Endpunkt dafür verantwortlich, ein Datenpaket zu empfangen, das von dem Host gesendet wird, und in einem Puffer zu speichern. Wenn der Endpunktkanal momentan blockiert ist, wird am Ende der Datenübertragung ein STALL-Handshakepaket gesendet. Wenn  
55 der Endpunktkanal momentan disabled ist, wird am Ende der Datenübertragung kein Handshakepaket gesendet. Wenn der Endpunktkanal enabled ist, jedoch kein Puffer vorhanden ist, in dem die Daten gespeichert werden können, wird ein NAK- (negative Bestätigung) Handshakepaket gesendet.

- Ein disabled Endpunkt oder Endpunkte, die momentan nicht auf einen Endpunktkanal abgebildet sind, reagieren  
60 nicht auf IN-, OUT- oder SETUP-Token.

- Der EPC bewahrt getrennte Zustandsinformation für jeden Endpunktkanal auf. Die Zustandsinformation, die für unmittelbare Entscheidungen erforderlich ist (z. B. für das Antworten mit einer NAK oder STALL-Quittung) wird zu allen Zeiten für jeden Endpunkt in dem Modul bereitgehalten. Andere Information, die für den Endpunktbetrieb notwendig ist, wird in dem Endpunktfeld (Array) aufbewahrt. Da zu einer gegebenen Zeit nur ein Endpunktkanal aktiv ist, wird die  
65 Endpunktkanal-Ablaufsteuereinheit (state machine) von allen Endpunktkanälen gemeinsam benutzt.

Bei IN-Token ist der EPC für die Übertragung der Daten von dem definierten Puffer zum Host verantwortlich. Bei OUT-Token ist der EPC für die Übertragung von Daten von dem Host zu dem definierten Puffer verantwortlich. Um eine größere Flexibilität zu ermöglichen, wenn Puffer einem Endpunktkanal zugewiesen werden, ist für jeden Endpunktkanal

# DE 199 00 369 A 1

ein Doppelpufferverfahren zulässig. Dadurch kann der EPC eine schnelle Antwort auf alle Transaktionen an dieses Modul liefern, ohne auf NAKs zurückzugreifen.

## Abbildung auf Endpunktkanäle

Der EPC sieht die Abbildung der Funktionsadresse, Tokentyp (IN, OUT, SETUP) und Endpunktnummer auf einen der Endpunktkanäle vor. Zwei feste Abbildungen auf Endpunktkanäle werden unterstützt und von den Einstellungen des Endpunktkanal-Abbildungsauswahl-Bits des Einrichtungs-Konfigurationsregisters (DCR.EPMS) gesteuert.

Wenn EPMS = 0:

Funktionsadresse	Tokentyp	Endpunkt	Endpunkt-Kanalnummer
FA0	IN,OUT,SETUP	0	0
FA0	IN,OUT,SETUP	1	1
FA0	IN,OUT,SETUP	2	2
FA0	IN,OUT,SETUP	3	3
FA0	IN,OUT,SETUP	4	4
FA0	IN,OUT,SETUP	5	5
FA0	IN,OUT,SETUP	6	6
FA0	IN,OUT,SETUP	7	7
FA0	IN,OUT,SETUP	8	8
FA0	IN,OUT,SETUP	9	9
FA0	IN,OUT,SETUP	10	10
FA0	IN,OUT,SETUP	11	11
FA0	IN,OUT,SETUP	12	12
FA0	IN,OUT,SETUP	13	13
FA0	IN,OUT,SETUP	14	14
FA0	IN,OUT,SETUP	15	15
Nicht anwendbar	SOF	Nicht anw.	SOF-Kanal
Nicht FA0	beliebig	Nicht anw.	keine

Wenn EPMS = 1:

Funktionsadresse	Tokentyp	Endpunkt	Endpunkt-Kanalnummer
FA0	IN,OUT,SETUP	0	0
FA0	IN,OUT,SETUP	8	1
FA0	OUT	1	2
FA0	IN	1	3
FA0	OUT	2	4
FA0	IN	2	5
FA0	OUT	3	6
FA0	IN	3	7
FA0	OUT	4	8
FA0	IN	4	9
FA0	OUT	5	10
FA0	IN	5	11
FA0	OUT	6	12
FA0	IN	6	13
FA0	OUT	7	14
FA0	IN	7	15
FA0	SETUP	1-7	Keine
FA0	Beliebig	9-15	Keine
Nicht anwendbar	SOF	Nicht anw.-	SOF-Kanal
Nicht FA0	Beliebig	Nicht anw.-	Keine

## Endpunktsteuerung

Alle von den Endpunktkanälen während der Datenübertragungen verwendete Information wird in dem Endpunktfeld aufbewahrt, das im RAM liegt. Dieses Feld ist durch den Endpunktkanal auf der Basis indexiert, die in dem Endpunktfeld-Basisadressen-Register (EABA) definiert ist. Dieses Feld enthält zwei Doppelworte pro Endpunkt. Jedes Doppelwort ist ein Steuerwort, das Steuer- und Statusinformation für den Endpunktkanal enthält, wie einen Zeiger zum Anfang des Puffers, die Größe des Puffers, den Zustand des Puffers und andere relevante Parameter.

Der Pufferzustand definiert, ob der Puffer für Eingangs- oder Ausgangstransaktionen verwendet werden soll und ob auf ihn momentan zugegriffen wird oder ob er die gewünschte Operation beendet hat.

Wenn eine IN- oder OUT-Übertragung von einem Endpunktkanal angefordert wird und ein Puffer vorhanden ist, wird

das geeignete Steuerwort gelesen. Dieses Steuerwort teilt dem Endpunkt-Controller mit, wie die Transaktion zu beenden ist. Am Ende einer Transaktion wird in das Steuerwort der Status geschrieben, der von der Software als Anzeige zum Freigeben dieses Puffers verwendet wird, und ein neues wird vorgesehen.

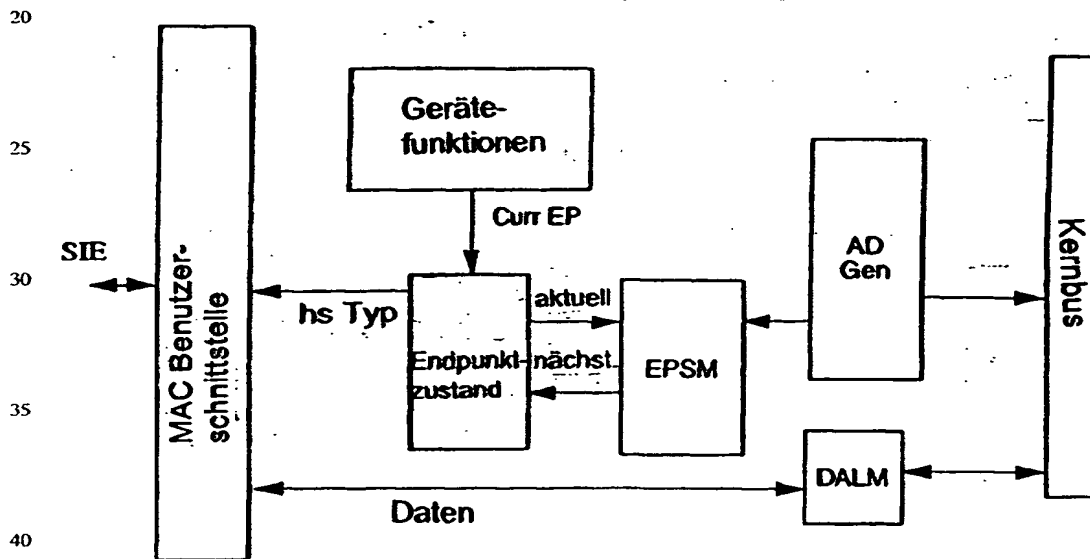
#### Endpunkt-Controller-Blockdiagramm

Der Endpunkt-Controller besteht aus mehreren Funktionsblöcken. Die Endpunkt-Ablaufsteuereinrichtung (state machine) sieht die zentralisierte Steuerung für die Dienstblöcke des Datenausricht-Mux (DALM; data alignment max), Adressengenerators (ADGen), Teilendpunktkanal-Speichers (PEP-Speichers) vor. Die Gerätefunktion, wie die Adreßüberprüfung und -erzeugung werden in dem Gerätefunktionsblock realisiert.

Alle Ablaufsteuereinrichtungen laufen mit dem USB-Takt von 12 MHz. Alle Datenübertragungen arbeiten mit dem Kernbus-Takt. Teile des ADGen-Blocks und der gesamte DALM-Block arbeiten mit dem Kernbus-Takt. Alle anderen Blöcke des Endpunkt-Controllers verwenden ausschließlich den lokalen Takt von 12 MHz. Datenbytes zwischen den Bereichen werden bei der MAC-Benutzerschnittstelle mit einem asynchronen Handshake für jedes übertragene Byte synchronisiert.

Fig. 2-2

Vereinfachtes Blockdiagramm des Endpunkt-Controllers



#### Speicherzugriffsleistung

Der Endpunktcontroller minimiert die Anzahl der erforderlichen Speicherzugriffe. Nur bei Empfang eines Tokens ist ein Speicherzugriff auf das Endpunktfeld notwendig. Ein einzelner Doppelwort-Zugriff, zusammen mit dem systemeigenen Endpunktzustand, ist ausreichend, damit der Endpunktkanal die Hostanfrage verarbeiten kann. Dieser Zugriff liest die Positions- und Größeninformation für die Datenübertragung. Die Datenübertragung erfolgt in aufeinanderfolgenden Doppelwort-Lese- oder Schreib-Operationen.

Das Modul hat die höchste Priorität auf dem Kernbus für Übertragungen, und die Übertragungsrates ist deutlich höher als die des USB. Es besteht somit eine minimale Notwendigkeit der Datenpufferung innerhalb des Endpunkt-Controllers, abgesehen von einem extra Wort, um nachfolgende Zugriffe mit dem aktuellen Zugriff überlappen zu lassen. Ein Byte-zähler überwacht die Anzahl der Bytes, die übertragen werden. Der letzte Zugriff einer Übertragung kann ein teilweises Lesen oder Schreiben des Speichers erfordern. Das teilweise Schreiben des letzten Zugriffs wird von dem letzten Byte-zählwert bestimmt. Am Ende einer Übertragung wird der Status für die Übertragung in das aktuelle Steuerwort geschrieben, und das andere Steuerwort für diesen Endpunkt wird gelesen, um zu überprüfen, ob ein weiterer Puffer vorbereitet wurde. Nach den Steuerwort-Operationen wird der Zustand des Endpunktkanals aktualisiert.

#### Synchronisationssupport

Durch eine Kombination aus Hardware und Software unterstützt das UNIM alle drei USB-definierten Synchronisationsklassen, asynchron, synchron und adaptiv. Der Endpunkt-Controller sieht eine Hardwareunterstützung für die Überwachung des USB-Bustaktes von 1 kHz im Verhältnis zu einem externen Bezugstakt, von z. B. 8 kHz, und zum Überwachen der Qualität und Konsistenz des USB-Bustaktes von 1 kHz vor.

Steigende Flanken eines externen Bezugstaktes werden in dem Bezugs-Zählregister gezählt. Es kann dazu verwendet werden, die USB-Blocknummer zu vergleichen. Die Frequenzdifferenz zwischen dem Bezugstakt und dem USB-Takt kann dem Host in Paketen signalisiert werden, so daß der Host die USB-Bustaktfrequenz einstellen kann; somit kann der USB-Takt mit dem Bezugstakt synchronisiert werden.

# DE 199 00 369 A 1

Der Endpunkt-Controller hält auch einen Status aufrecht, um anzuzeigen, wenn er zu dem USB-Bustakt von 1 kHz synchronisiert ist und wenn aufgrund eines fehlenden oder falschen Block-Anfang-Pakets eine Diskontinuität in dem USB-Bustakt auftritt. Er wird von der Software dazu verwendet, die Warteschlange der isochronen Pakete einzustellen. Die aktuelle Blocknummer wird von dem Endpunkt-Controller dazu verwendet, zu ermitteln, welcher isochrone Puffer als nächstes gesendet werden soll.

## Betrieb aussetzen/wieder aufnehmen

### Aussetzbetrieb

Es wird erwartet, daß eine USB-Einrichtung abhängig von dem Aussetzereignis, das auftritt, nachdem 3 ms verstrichen sind, ohne daß irgendwelche Busaktivität erfaßt wurde, in den Aussetzzustand geht. Das UNIM erfaßt dieses Ereignis und zeigt es dem CR32A an, indem das ISR.SD Bit gesetzt und ein Interrupt gesendet wird, wenn es enablt ist. Das CR32A sollte abhängig davon das UNIM in den Aussetzzustand versetzen und nach 5 ms ohne Erfassung irgendwelcher Aktivität in den Haltemodus umschalten (ISR.SD5).

Im Aussetzzustand geht der Transceiver in einen speziellen Niederleistungsmodus, und die Takte können unterbrochen werden. Der gespeicherte Zustand und das Kernbus-RAM bleiben statisch, so daß bei der Wiederaufnahme keine weiteren Operationen notwendig sind.

Normandy kann aufgrund eines für den CR32A lokalen Ereignisses den Betrieb nach dem Aussetzzustand wieder aufnehmen, wodurch seinerseits der USB-Bus über eine Fern-Wiederaufnahme oder bei Erfassung eines Fern-Befehls auf dem USB-Bus, der den CR32A über die MIWU unterbricht, geweckt werden kann.

Man beachte, daß eine Einrichtung im Aussetzbetrieb maximal 500 µA von dem Bus ziehen darf, siehe "GERÄTE-SPEZIFIKATIONEN" auf Seite 3-55 für Einzelheiten des Stromverbrauchs von Normandy.

### Fernwiederaufnahme

Wenn der Host Fernweckrufe von diesem Knoten enablt hat, sendet Normandy einen solchen abhängig von einem lokalen Ereignis, wie einem Interrupt.

Wenn die Firmware das Ereignis erfaßt, nimmt sie die Prozessoraktivität im aktiven Modus wieder auf. Sie weckt dann das UNIM aus dem Niederleistungs-Aussetzzustand und löst eine Fernwiederaufnahme auf dem USB unter Verwendung des FSR-Registers aus. Die Knotenfirmware muß sicherstellen, daß wenigstens 5 ms Leerlauf auf dem USB stattfinden. Im Wiederaufnahmestand wird auf dem USB eine Konstante "K" angezeigt. Diese sollte wenigstens 1 ms andauern, wobei danach der USB-Host damit fortfährt, das Wiederaufnahmesignal während wenigstens zusätzlicher 20 ms zu senden, und dann die Wiederaufnahmeoperation abschließt, indem er die Paketende-Sequenz ausgibt. Sollte innerhalb von 100 ms kein Paketende-Signal vom Host empfangen werden, sollte die Fernwiederaufnahme erneut durch Software gestartet werden.

### USB-Wiederaufnahmeoperation

Bei Erfassung eines Wiederaufnahme- oder Reset-Signals während des Aussetzzustands ist das UNIM-Modul dafür verantwortlich, Normandy aufzuwecken. Dies erfolgt, indem das UNIM ein Signal (ohne Beitrag irgendwelcher Takte) an die MIWU sendet, das die Erfassung einer Wiederaufnahme auf dem USB anzeigt und den Normandy-Weckruf anfordert, um den CR32A zu aktivieren und zu unterbrechen. Dabei werden die Takte für das USB-Modul erneut enablt, und der Rest des Chips wird aufgeweckt.

Die USB Spezifikation erfordert, daß eine Einrichtung innerhalb von 10 ms auf USB-Token antworten kann.

### Einschalt-Resets

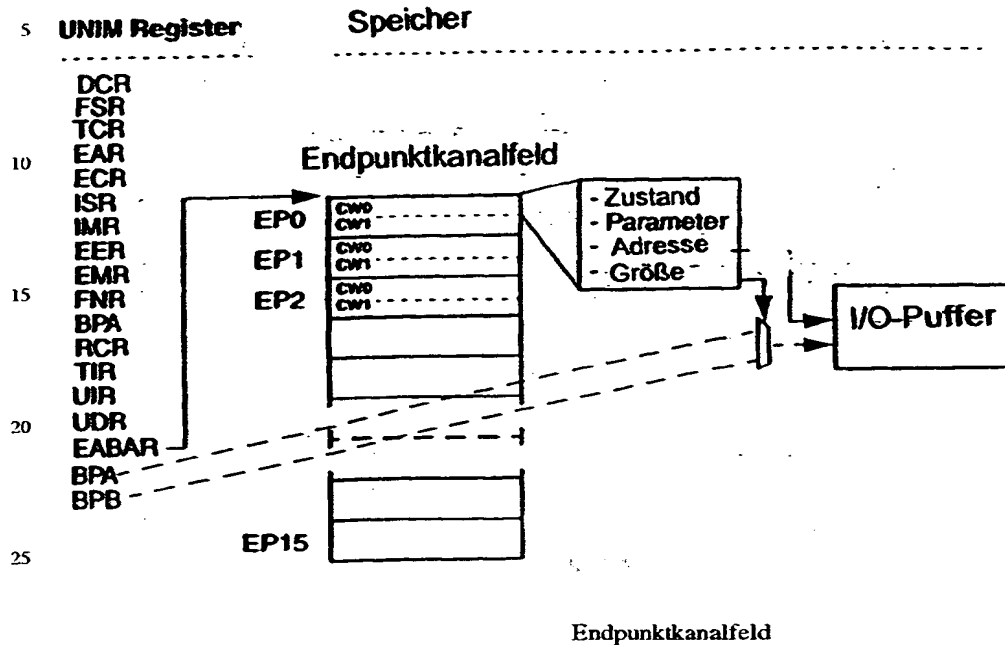
Es gibt zwei Fälle, in denen dieser Knoten aktiviert werden kann; wenn der Knoten in einen bereits aktivierten USB-Verteilerport eingesteckt wird, und wenn bei dem Verteilerknoten die Energie eingeschaltet wird. Um eine Interoperabilität zwischen den Bus-Geräten zu erreichen, erfordert die USB-Spezifikation, daß der Knoten innerhalb von 100 ms auf einen USB-Reset reagieren können muß, nachdem die Energie eingeschaltet und beim Verteilerport stabil ist. Ähnlich muß ein Knoten, der in Betrieb eingesteckt wird, auf ein USB-Reset innerhalb von 100 ms reagieren, nachdem seine Verbindung von einem Verteilerport erfaßt wurde. Der USB-Reset wird während wenigstens 10 ms beibehalten. Nachdem der USB-Reset gelöst wurde, liefert der Verteiler 100 mA Busleistung an das Gerät. Erst nachdem das Gerät konfiguriert wurde, darf es die in der USB-Standardkonfiguration beschriebene Energiemenge aufnehmen (bis zu 500 mA). Das UNIM führt eine USB-definierte, Bus-getriebene Hochleistungs-Funktion aus.

### 2.1.6 UNIM Programmiermodell

Das USB-Modell enthält Register, die dazu verwendet werden, den Betrieb der USB-Einrichtung und ein Endpunktfeld im Speicher zu steuern und zu überwachen, das zum Steuern der Endpunktparameter verwendet wird, die für eine bestimmte Endpunktübertragung spezifisch sind. Die Beziehung zwischen den verschiedenen Registern und Speicherparametern ist unten angegeben.

Fig. 2-3

USB-Modul-Programmiermodell



Das Endpunktkanalfeld besteht aus 32 Steuerworten, 2 für jeden Endpunktkanal. Das Endpunktkanalfeld liegt im Speicher bei einer Adresse, auf die EABAR weist. Diese Adresse muß eine ausgerichtete 128-Byte Adresse sein (d. h. 7 LSBs der Adresse sind 0). Das Endpunktkanalfeld erhält über die 4-Bit Endkanalnummer einen Index, und das Steuerwort, auf das zugegriffen wird, wird von dem 1-Bit Zeiger des Endpunktkanal-Steuerwortes bestimmt. Zusammen erzeugen das 250-Bit EABAR, das 4-Bit EP und das 1-Bit CW eine 32-Bit Adresse, die zu einer Doppelwort-(32-Bit)Grenze ausgerichtet ist, und sie werden dazu verwendet, auf das 32-Bit Steuerwort zuzugreifen. Das Format jedes Eintrags in der Tabelle ist unten beschrieben.

Steuerwortformat

Das 32-Bit Steuerwort enthält die gesamte Information, die zum Betreiben dieses Endpunktkanals notwendig ist, wenn er aktiv ist. Es gibt zwei solche Einträge für jeden Endpunktkanal. Das Steuerwort, auf das als nächstes zugegriffen werden soll, wird von dem Wert des Endpunktkanal-Steuerwortzeigers (CWP) bestimmt. Unmittelbar nach dem Empfang eines Tokens IN, OUT oder SETUP durch einen Endpunktkanal, der momentan in einem betriebsbereiten Zustand ist, wird das Steuerwort des ausgewählten Endpunktkanals bei dem aktuellen Steuerwortzeiger gelesen, um zu ermitteln, wie auf den Token reagiert werden soll. Am Ende einer Übertragung, wenn ein Beendigungsstatus geschrieben wird, wird das Steuerwort mit dem aktuellen Status aktualisiert, der CWP wird inkrementiert, und das nächste Steuerwort wird gelesen. Wenn ein Zwischenstatus geschrieben wird, wird nur das höchstwertige Byte des Steuerwortes aktualisiert.

3	1	8	27	26	16	15	14	13	12	11	10	9	0
BS		res		BA	BP	BR	TE	TI	IC	IE		BC	

BS Pufferzustand

4-Bit-Wert. Von Software und EPC aktualisiert, um die Steuerung dieses Puffers zwischen ihnen zu synchronisieren. Der EPC schreibt den Status, der die Verwendung des EPCs angibt. Dieses Feld wird von der Knoten-Firmware auf Ausgang bereit (Output Ready), Eingang bereit (Input Ready) Setup bereit (Setup Ready), Überspringen (Skip), Unterbrechen (Stall) und Disable eingestellt. Alle anderen Werte werden von dem EPC eingestellt. Der EPC kann auch die Werte von Stall und Disable bei Fehlerbedingungen schreiben. Der Puffer kann von der Knoten-Firmware wieder beansprucht werden, wenn der Pufferzustand den Wert eines Fertig-Status hat; dies umfaßt den Eingabe/Ausgabe-Fertigstatus und die Fehler Codes.

0: Skip Skip. Wird zusammen mit dem Ausgabe-Fertigbefehl verwendet. Skip bewirkt, daß das nächste empfangene

Datenpaket ignoriert und der Steuerwortzeiger inkrementiert wird. Bei Steuerendpunkten bewirkt es auch, daß der Endpunktkanalzustand von dem Eingabe/Ausgabe-Bereitzustand auf den Eingabe/Ausgabe-Wartezustand schaltet. Das Einstellen von TE und TI bestimmt, welche Quittung (Handshake) zurückgegeben wird, sowie den nächsten Zustand des Endpunktkanals. NAK wird zurückgegeben, wenn entweder TE oder TI gesetzt ist. Wenn TI gesetzt ist und nicht TE, wird keine Quittung zurückgegeben. Wenn TE gesetzt ist, wird ACK zurückgegeben, und ein vollständiger Status wird geschrieben, wenn ein Paket mit der richtigen PID empfangen wird. 5

1: IRDY Input Ready (Eingang bereit). In den Puffer wurden Daten geschrieben, die abhängig von einem IN-Token gesendet werden sollen.

2: ORDY Output Ready (Ausgang bereit). Der Puffer wurde für das Schreiben zugewiesen.

3: SRDY Setup Ready. Der Puffer wurde für das Schreiben zugewiesen. Dies wird nur während der ersten Übertragung von Steuerlese- und Schreiboperationen bei einem Steuerendpunkt verwendet. Es wird nur in das Steuerwort 0 geschrieben. 10

4: OWT Output Wait (Ausgabe warten). Gibt an, daß ein Fehler aufgetreten ist, während OUT-Daten von dem Host empfangen wurden. Dies umfaßt die Erfassung eines Bit-Stuffing-Fehlers. Ein erneuter Versuch wird erwartet, weil dies als Zwischenzustand betrachtet wird, außer in dem Fall eines isochronen OUT-Endpunkts, wo es als Beendigungsstatus behandelt wird. 15

5: CRCE CRC Error. Gibt an, daß ein fehlerhafter CRC am Ende einer Ausgabetransaktion erfaßt wurde. Ein erneuter Versuch wird erwartet, weil dies als Zwischenzustand betrachtet wird, außer in dem Fall eines isochronen OUT-Endpunkts, wo es als ein Beendigungsstatus behandelt wird.

6: BUFE Buffer Error (Pufferfehler). Während einer Ausgabetransaktion, wurden Daten von dem USB schneller empfangen, als sie in den Speicher geschrieben werden konnten. Während einer Eingabe wurden Daten aus dem Speicher nicht schnell genug gelesen, um mit der USB-Datenrate mitzuhalten. An diesem Endpunkt wird ein erneuter Versuch erwartet. 20

7: IWT Input Wait (Eingabe warten). Der Puffer wird von dem Endpunkt-Controller angehalten und wartet darauf, diesen Puffer bei der nächsten Gelegenheit zu senden oder erneut zu senden. Dies gilt für den Fall einer fehlenden Bestätigung (ACK). 25

8: ICMP Input Complete (Eingabe beendet). Erfolgreiche Beendigung der Eingabe-Operation. Der Puffer kann freigegeben werden.

9: OCOMP Output Complete (Ausgabe beendet). Der Puffer ist mit Daten gefüllt, und es wurde kein Fehler erfaßt.

10: TKNE Token Error. Ein Token des falschen Typs für den momentanen Kontext des Steuerendpunktkanals wurde empfangen. Wenn es ein Ausgangspuffer war, wird der Endpunkt angehalten. Wenn es ein Eingangspuffer war, wird NAK gesendet, und der Zustand des anderen Steuerworts wird untersucht. Dies wird als ein Beendigungsstatus angesehen. 30

11: SYNE Iso Synchronisation Error. Daten wurden in einem Puffer mit TE = 1 und TI = 1 empfangen, der CWP war jedoch nicht gleich dem LSB des FNR. Dies wird als ein Beendigungsstatus angesehen. 35

12: DATE Data Error (Datenfehler). Für OUT-Daten, wenn CW.BR nicht gesendet wurde und die Menge der vom Host empfangenen Daten geringer als erwartet war oder der zugewiesene Puffer war nicht groß genug, um die OUT-Daten zu speichern. Für Eingangsdaten wird es gesetzt, wenn ein Pufferfehler aufgetreten ist und CW.TE nicht gesetzt oder CW.TI gesetzt war. Dies wird als ein Beendigungsstatus angesehen.

13: rsvd Reserviert. 40

14: HALT Halt. Bewirkt, daß der aktuelle Endpunktkanal in seinen Halt-Zustand geht, oder gibt an, daß der Endpunkt in den Halt-Zustand gegangen ist.

15: DIS Disable. Bewirkt, daß der aktuelle Endpunktkanal in seinen Disable-Zustand geht, oder zeigt an, daß der Endpunktkanal in den Disable-Zustand gegangen ist.

TI: Type isochron. 1 isochron, 0 Steuerung, Bulk, Interrupt. 45

TE: Toggle Enable. Wird zum Vergleichen/Erzeugen des Wertes des Datenpaket-Identifikators (PID) (Data0/Data1) mit dem Steuerwortzeiger verwendet.

0: Der Daten-Toggle-PID wird bei Ausgangstransaktionen nicht überprüft, und er wird gestützt auf den aktuellen Wert des CWP abhängig von IN-Token erzeugt. Der CWP wird nach jeder Transaktion inkrementiert (invertiert), selbst wenn kein ACK empfangen wurde. Bei isochronen Endpunkten werden Ausgangsdaten nicht in den Speicher geschrieben, bis der CWP gleich dem niedrigstwertigsten Bit der Blocknummer FNR.FN(0) ist. Dies hat keinen Einfluß auf Daten, die abhängig von IN-Token gesendet werden. Daten-Toggle sollte für Puffer nicht enablt werden, die dafür bestimmt sind, mit dem SETUP-Token gesendete Daten zu empfangen. 50

1: Der Daten-Toggle-PID wird bei Ausgangstransaktionen überprüft und gestützt auf den aktuellen Wert von CWP abhängig von IN-Token erzeugt. Bei Ausgangstransaktionen wird ein ACK abhängig von Ausgangspaketen gesendet, deren PID-Wert (Data 0/1) nicht mit dem CWP übereinstimmt. Bei Eingangstransaktionen wird CWP nach jeder Transaktion nur inkrementiert (invertiert), wenn eine ACK-Quittung abhängig von einer Zeitablaupperiode zurückgegeben wurde. Bei isochronen Endpunkten werden die Ausgangsdaten immer in den Speicher geschrieben, wenn jedoch FNR.FN(0) nicht gleich CWP ist, wird der ISO-Sync-Fehlerpufferzustand zurückgeschrieben.

BR Buffer Rounding (Pufferrundung). Gibt für Ausgangstransaktionen an, ob es zulässig ist, einen Puffer mit einer anderen Größe als der des empfangenen Datenpakets zu verwenden. 60

0: Das Datenpaket sollte den definierten Datenpuffer genau füllen. Der Pufferüberlauf- oder Unterlaufstatus wird zurückgegeben, wenn das Datenpaket nicht genau in den Puffer paßt.

1: Das Datenpaket kann kleiner als der definierte Puffer sein, ohne eine Fehlerbedingung am Endpunkt zu erzeugen. Wenn es zusammen mit einem Ausgabeende- oder Datenfehlerpuffer-Status gelesen wird, gibt dieses Feld den Wert des empfangenen PID an – 0 für Data0, 1 für Data1. 65

IC Interrupt on Packet Complete (bei Paketende). Dies bewirkt, daß an diesem Endpunkt ein Ereignis gesetzt wird, (EER.EPn), wenn der Pufferstatus mit Eingabeende oder Ausgabeende beschrieben wird, bei einem Paketfehler,

oder wenn der Endpunktkanal unterbrochen wird.

IE Interrupt on Packet Error (bei Paketfehler). Dies bewirkt, daß an diesem Endpunkt ein Ereignis gesetzt wird (EER.Epn), wenn der Pufferzustand einem der Pufferfehlercodes entspricht, oder wenn der Endpunkt unterbrochen wird. Wenn sowohl IC als auch IE gesetzt sind, bewirken auch CRC- und Bit-Stuffingfehler, daß ein Ereignis erzeugt wird und diese Puffer nicht mehr verwendet werden, wie im normalen Betrieb. Dies dient hauptsächlich der Diagnose.

res Reserviert. 1 Bit.

BP Buffer Page (Pufferseite).

0: Verwende Pufferseite A als obere Adresse des Eingabe/Ausgabepuffers.

1: Verwende Pufferseite B als obere Adresse des Eingabe/Ausgabepuffers.

BA Buffer Address (Pufferadresse). 11 Bit. Die unteren 11 Bit der Pufferadresse. Der Puffer muß bei einer ausgeordneten Doppelwortgrenze beginnen, und er kann bei jeder Byte-Grenze enden. Am Ende aller vollständigen Übertragungen wird der Restwert der Pufferadresse geschrieben. Am Ende unvollständiger Übertragungen, z. B. wenn der Status mit Eingabe- oder Ausgabe-Warten beschrieben wird, wird dieses Feld nicht überschrieben. Bei OUT-Puffern ist das erste gespeicherte Byte das Byte, daß dem PID folgt. Ähnlich zeigt bei IN-Puffern die Adresse auf das erste Byte der zu übertragenden Daten, d. h. das Byte, das dem PID in dem Paket unmittelbar folgt. Der CRC wird automatisch berechnet und an das Paket angehängt, außer wenn der IN-Token aus dem IS RDY-Zustand empfangen wird und das IS-Feld des Steuerworts nicht auf 1 gesetzt ist.

BC Byte Count. 10 Bit. Die Größe des zu verwendenden Datenpuffers. Der Puffer beginnt bei einer geraden Doppelwortgrenze, und er kann bei jeder Bytegrenze enden. Bei IN-Transaktionen zeigt dies die Größe der zu übertragenden Daten an. Bei OUT-Transaktionen gibt dies die Größe des Puffers an, in dem Daten empfangen werden können. 0 ist ein zulässiger Wert für den Byte-Zählwert. Am Ende aller vollständigen Übertragungen wird der Restwert des Bytezählers geschrieben. Am Ende unvollständiger Übertragungen, z. B. wenn der Zustand als Eingabe-Warten oder Ausgabe-Warten beschrieben wird, wird dieses Feld nicht überschrieben.

#### 2.1.7 Endpunktkanalbetrieb

##### Endpunktkanalzustände

Um den Betrieb und die Steuerung aufrechtzuerhalten, hält jeder Endpunktkanal einen Zustand mit genau definierten Bedingungen für Übertragungen zwischen den Zuständen aufrecht, die eine implizite Hardware/Software-Synchronisierung benötigen. Endpunktkanal-Zustandsübergänge treten auf:

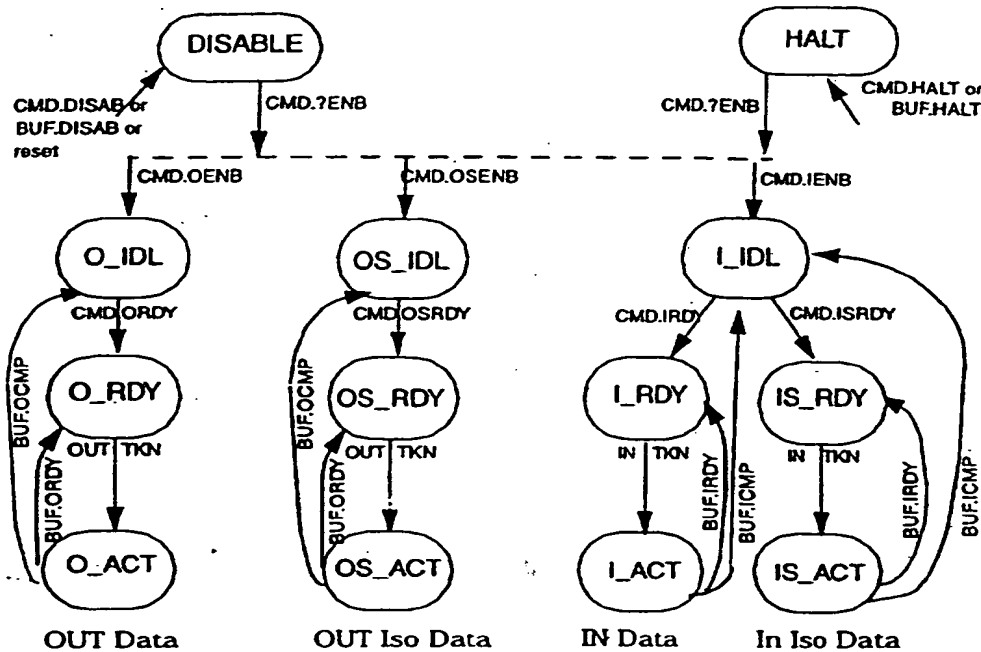
- bei expliziten Softwarebefehlen an einen Endpunkt über das Endpunktbefehlsregister (ECR)
- durch Lesen des Pufferstatus des anderen Puffers in dem anderen Steuerwort für diesen Endpunkt am Ende einer Übertragung, und
- als eine Folge von Ereignissen auf dem USB, wie dem Empfang von IN, OUT und SETUP-Token und ACK-Quittungen.

Fig. 2-4 zeigt eine vereinfachte Endpunktkanal-Ablaufsteuereinrichtung zum Verarbeiten von IN und OUT-Transaktionen. Die Verarbeitung von Steuer-Lese- und Schreibtransaktionen erfolgt auf ähnliche Weise.



Fig. 2-4

## Vereinfachtes Endpunktzustandsdiagramm



Der aktuelle Endpunktkanalzustand ist im Endpunktzustandsregister sichtbar, auf das über den USB-Index und Datenregister zugegriffen werden kann. Maximal ist immer ein Endpunktkanal zu einer gegebenen Zeit in einem der aktiven Zustände. Dies ermöglicht eine gemeinsame Nutzung der aktiven Zustandslogik durch alle Endpunktkanäle.

**Disable.** Der Disable-Zustand wird bei einem Reset, bei einem Disable-Befehl an diesen Endpunktkanal und bei Erfassung eines Disable-Zustands im Pufferzustand eines Steuerwortes ausgelöst. Jeder Token, der von diesem Endpunktkanal empfangen wird, während er in diesem Zustand ist, wird ohne Handshake zurückgegeben. Ein Eingreifen mittels Software ist notwendig, um den Endpunktkanal aus diesem Zustand zu holen, üblicherweise über einen Enable-Befehl an einen der Wartezustände.

**Halt.** Der Halt-Zustand wird aktiviert, wenn der Stall-Befehl (Unterbrechen) an den Endpunktkanal geht, wenn ein Stall-Zustand im Pufferzustand eines Steuerwortes erfaßt wird, sowie bei bestimmten Fehlerbedingungen. Jeder von diesem Endpunktkanal empfangene Token wird als Stall-Handshake zurückgegeben, während dieser in diesem Zustand ist, um anzuzeigen, daß der Endpunkt momentan unterbrochen ist. Ein Eingreifen mittels Software ist erforderlich, um den Endpunktkanal aus diesem Zustand zu bringen, üblicherweise über einen Enablebefehl an einen der Wartezustände.

**Input Idle I\_IDL.** In diesem Zustand erwartet der Endpunkt, daß bald ein Puffer zugewiesen wird. Ein NAK-Handshake wird zurückgegeben, wenn ein IN-Token auf diesen Endpunktkanal abgebildet wird. Keine Antwort wird bei Out-Token von diesem Endpunktkanal zurückgegeben.

**Input Ready I\_RDY.** Dieser Zustand wird über einen Eingang-Bereit-Befehl an diesen Endpunktkanal oder bei Erfassung eines anderen Eingabepuffers am Ende einer Übertragung aktiviert. Der Befehl sollte nur erfolgen, nachdem diesem Endpunkt ein Puffer zugewiesen wurde, indem das Steuerwort bei der richtigen Position in dem Endpunktfeld geschrieben wird. Diese Steuerwort sollte auf einen Datenpuffer zeigen, der Daten enthält, die abhängig von dem nächsten IN-Token, welcher von diesem Endpunktkanal empfangen wird, an den Host zurückgegeben werden sollen. Der PID-Wert wird als Data0 übertragen, wenn CWP gleich 0 ist, und als Data1, wenn CWP gleich 1 ist.

**Input Active I\_ACT.** Dieser Zustand wird bei Empfang eines IN-Tokens aktiviert, wenn der ausgewählte Endpunktzustand momentan Eingang-Bereit ist. In diesem Zustand werden Daten von dem Speicher an den USB übertragen. Bei Beendigung der Datenübertragung sollte ein ACK-Handshake innerhalb von 16 Bitzeiten empfangen werden. Am Ende dieser Zeitablaufperiode wird der Pufferwartezustand in das Steuerwort zurückgeschrieben, wenn kein ACK empfangen wurde. Der Adreß- und Bytezähler werden jedoch nicht überschrieben, weil eine erneute Übertragung erfolgt und die momentanen Werte nicht neu verwendet werden müssen. Wenn ein ACK empfangen wird, werden der Pufferstatus zusammen mit dem aktuellen Wert des Adreß- und Bytezählers geschrieben. Dann wird das andere Steuerwort an diesem Endpunkt gelesen, um den nächsten Zustand des Endpunktes zu ermitteln, der das nächste Mal verwendet werden soll, wenn ein Token an diesen Endpunktkanal gerichtet wird. Wenn dieser Endpunkt für eine Steuerübertragung verwendet wird, kann ein Übergang zu dem Ausgang-Bereit- oder Ausgang-Ruhe-Zustand erfolgen, um die Statusphase der Steuerleseübertragung zu beenden.

**Input ISO Ready IS\_RDY.** Dieser Zustand wird über einen Eingang-Bereit-Befehl an diesen Endpunktkanal oder bei Erfassung eines weiteren Eingabepuffers am Ende einer Übertragung aktiviert. Der Befehl sollte nur erfolgen, nachdem diesem Endpunkt ein Puffer zugewiesen wurde, indem das Steuerwort an die geeignete Stelle in dem Endpunktfeld geschrieben wird. Dieses Steuerwort sollte zu einem Datenpuffer weisen, der Daten enthält, die abhängig von dem nächsten, von diesem Endpunktkanal empfangenen IN-Token an den Host zurückgegeben werden sollen. Der PID-Wert wird unabhängig vom Wert des CWP immer als Data0 übertragen.

Input ISO Active IS\_ACT. Dieser Zustand wird bei Empfang eines IN-Tokens aktiviert, wenn der ausgewählte Endpunktzustand Eingang-momentan-bereit ist. Während dieses Zustands werden Daten vom Speicher zu dem USB übertragen. Bei Beendigung der Datenübertragung werden der Pufferstatus zusammen mit dem aktuellen Wert des Adreß- und Bytezählers am Ende der Übertragung geschrieben. Dann wird das andere Steuerwort bei diesem Endpunkt gelesen, um den nächsten Zustand des Endpunktes zu bestimmen, der das nächste Mal verwendet werden soll, wenn ein Token an diesen Endpunktkanal gerichtet wird.

Output Idle O\_IDL. In diesem Zustand erwartet der Endpunkt, daß bald ein Puffer zugewiesen wird. Ein NAK-Handshake wird zurückgegeben, wenn ein OUT-Token auf diesen Endpunktkanal abgebildet wird. Keine Antwort wird zurückgegeben, wenn IN-Token an diesen Endpunktkanal gehen.

Output Ready O\_RDY. Dieser Zustand wird aktiviert, nachdem dem Endpunktkanal ein Puffer zugewiesen wurde, in den Daten geschrieben werden sollen. Er kann über einen Ausgangs-Bereit-Befehl von dem O\_IDL-Zustand aktiviert werden, nachdem das passende Wort aus dem Endpunktfeld geschrieben wurde, oder bei Erfassung eines weiteren Ausgabepuffers am Ende einer Übertragung.

Output Active O\_ACT. Dieser Zustand wird bei Empfang eines OUT-Tokens aktiviert. Während dieses Zustandes werden Daten von dem USB in den Speicher geschrieben. Am Ende einer Ausgangsübertragung wird der Status in das Steuerwort bei dem CWP geschrieben. Wenn die Übertragung nicht erfolgreich war, d. h. wenn ein CRC oder Bitstuffing-Fehler erfaßt wurde, wird dies als Pufferstatus geschrieben. Wenn die Übertragung erfolgreich war, wird der Ausgabe-Beendet-Pufferstatus geschrieben, CWP wird invertiert, und das andere Steuerwort für diesen Endpunktkanal wird gelesen. Gestützt auf den Pufferzustand erfolgt der richtige Übergang. Wenn dieser Endpunkt für eine Steuerübertragung verwendet wird, kann ein Übergang zu dem Eingang-Bereit- oder zu dem Eingang-Warte-Zustand erfolgen, um die Statusphase einer Steuer-Schreibübertragung zu beenden.

Output ISO Idle OS\_IDL. In diesem Zustand erwartet der Endpunkt, daß bald ein Puffer zugewiesen wird. Kein Handshake wird zurückgegeben, wenn ein OUT-Token auf diesen Endpunkt ausgebildet wird. Keine Daten oder Handshakes wird bei IN-Token auf diesem Endpunktkanal zurückgegeben.

Output ISO Ready OS\_RDY. Dieser Zustand wird aktiviert, nachdem diesem Endpunktkanal ein Puffer zugewiesen wurde, in den Daten geschrieben werden sollen. Er kann über einen OS-Bereit-Befehl von dem OS\_IDL-Zustand aktiviert werden, nachdem das entsprechende Wort aus dem Endpunktfeld geschrieben wurde, oder bei Erfassung eines weiteren Ausgabepuffers am Ende einer Übertragung.

Output ISO Active OS\_ACT. Dieser Zustand wird bei Empfang eines OUT-Tokens von dem OS\_RDY-Zustand aktiviert. Während dieses Zustandes werden Daten in dem USB in den Speicher geschrieben. Am Ende einer Ausgangsübertragung wird der Status bei dem CWP in das Steuerwort geschrieben. Wenn die Übertragung nicht erfolgreich war, d. h. wenn ein CRC oder Bitstuffing-Fehler erfaßt wurde, wird dies als der Pufferstatus geschrieben. Wenn die Übertragung erfolgreich war, wird der Ausgabe-Beendet-Pufferstatus geschrieben. Am Ende der Statusübertragung wird CWP invertiert, und das andere Steuerwort für diesen Endpunktkanal wird gelesen. Der richtige Übergang erfolgt gestützt auf den Pufferzustand.

Setup Ready S\_RDY. Dieser Zustand wird aktiviert, nachdem diesem Endpunktkanal ein Puffer zugewiesen wurde, in den Daten geschrieben werden sollen. Dieser Zustand wird über einen Setup-Bereit-Befehl aktiviert, nachdem das richtige Wort aus dem Endpunktfeld geschrieben wurde, oder bei Erfassung des Setup-Bereit-Pufferzustands.

Setup Active S\_ACT. Dieser Zustand wird bei Empfang eines Setup-Tokens von dem Setup-Bereit-Zustand aktiviert. In diesem Zustand werden Daten von dem USB in den Speicher geschrieben. Am Ende einer Ausgangsübertragung wird der Status bei CWP in das Steuerwort geschrieben, CWP wird invertiert, und das andere Steuerwort für diesen Endpunktkanal wird gelesen. Der richtige Übergang erfolgt gestützt auf den Pufferzustand zu Ausgang-Warten, Ausgang-Bereit, Eingang-Warten oder Eingang-Bereit, abhängig vom Typ der Steuerübertragung und des gelesenen Pufferzustands.

#### Hardware/Software-Synchronisierung

Nach dem Reset werden alle Endpunkte disabelt. Ein Endpunkt wird enabelt, indem ein Enable-Befehl an den Endpunkt geschickt wird. Dies sollte nur erfolgen, wenn die Eingabe- und/oder Ausgabepuffer diesem Endpunkt bald zugewiesen werden. Alle Endpunkte, die momentan nicht auf einem Endpunktkanal abgebildet sind, werden als disabelt angesehen.

Die Software weist den Endpunktkanälen Puffer nach Bedarf zu. Nachdem jeder Puffer zugewiesen ist, führt er den Eingang- oder Ausgang-Bereit-Befehl an dem Endpunktkanal durch. Wenn der Endpunktkanal momentan im Ruhezustand ist, geht er in den geeigneten Bereit-Zustand. Wenn der Endpunktkanal momentan in einem Bereit- oder Aktiv-Zustand ist, ignoriert er alle Befehle außer den Disable- und Stall-Befehlen an den Endpunktkanal, die er sofort verarbeitet. Die Software muß diese Schreiboperation durchführen, um mögliche Synchronisierungsprobleme zu verhindern, weil die Behandlung des Endpunktkanals durch das UNIM zu den Softwareoperationen asynchron erfolgt.

Da jedem Endpunkt bis zu zwei Puffer zugewiesen werden können, kann der nächste Puffer vorbereitet werden, während der aktuelle abgearbeitet wird, um die Datenübertragung zu beschleunigen und unnötige NAK-Handshakes zu vermeiden.

#### 2.1.8 UNIM-Register

Beim Lesen sollten Daten aus reservierten Registern ignoriert werden, und beim Schreiben sollten diese Bits auf 0 geschrieben werden. – Dieser Vermerk ist in dem Chip global gültig.

#### Gerätesteuerregister (DCR)

Ein ein Wort breites Lese/Schreibregister. Das Gerätesteuerregister steuert die Einschaltsequenz des USB-Knotens.

# DE 199 00 369 A 1

Bei einem Reset wird dieses Register gelöscht. Das DCR-Registerformat ist unten angegeben:

15	14	13	12	11-9	8	7-5	4	3-2	1	0
res-DIAG	SFM	res_	ETE	res bei anderen EPMS	EPMS	res bei DEF3-1	DEF0	res ECE	NAT	

NAT. Node Attached (Knoten angeschlossen). Gibt an, daß dieser Knoten bereit ist und als an den USB angeschlossen erfaßt werden kann. Wenn es nicht gesetzt ist, erzwingt der Transceiver, daß das SE0 an dem USB-Port verhindert, daß der Verteiler (Hub), mit dem dieser Knoten verbunden ist, eine Verbindung erfaßt. Nach einem Reset ist es nicht gesetzt, so daß das Gerät Zeit hat, bevor es auf Befehle reagieren muß. Nachdem dieses Bit gesetzt ist, steuert das Gerät den USB nicht mehr an und sollte bereit sein, Reset-Befehle von dem Verteiler zu empfangen.

ECE. Endpunkt Controller Enable. Enabelt den Betrieb der Endpunktkanäle des Endpunktcontrollers, der die Endpunktfeld-Basisadresse und die Einstellung des Kanalabbildungs-Auswahlregisters verwendet. Wenn nicht gesetzt, werden alle Endpunkte als disabelt angenommen.

DEF 0-3. Default Address (Voreinstellungsadresse) 0-3. Enabelt die Voreinstellungsadresse und ignoriert den aktuellen Wert von FA. Dieses Bit wird gelöscht, wenn das entsprechende Adreß-Enable-Bit (AE) gesetzt wird, wenn das nächste Mal ein ACK abhängig von dem nächsten IN-Token empfangen wird, der seinerseits von einem Endpunktkanal empfangen wird, der für Steueroperationen verwendet wird. Wenn kein ACK innerhalb der Zeitablaufperiode empfangen wird, wird das nächste Mal auch dieses Bit zurückgesetzt, wenn ein Setup unmittelbar vor dem neuen Adreßfeld empfangen wird.

Man beachte, daß dann, wenn mehr als eine Adresse unterstützt wird, das niedrigstwertige DEF-Bit, für das das entsprechende AE-Bit gesetzt ist, das DEF-Bit ist, das beim nächsten Eintritt in den aktiven Eingabezustand auf einem Steuerendpunktkanal gelöscht wird. Es wird empfohlen, daß dann, wenn mehrere Funktionsadressen zugewiesen werden, dies von der niedrigsten zur höchsten Zahl erfolgt.

EPMS. Endpoint Pipe Map Select (Endpunktkanal-Abbildungsauswahl). Wählt die aktuelle Einstellung der Kanalabbildungstabelle aus. Eine Funktionsadreßübereinstimmung wird als gültig betrachtet, wenn das entsprechende AE-Bit gesetzt ist und das entsprechende DEF-Bit nicht gesetzt ist und die empfangene Adresse mit dem FA übereinstimmt. Wenn das DEF-Bit gesetzt ist, wird eine Übereinstimmung angenommen, wenn die 0-Adresse empfangen wird. Sollte mehr als eine Adresse realisiert sein, hat im Falle, daß mehrere DEF-Bits gesetzt sind, die niedrigste Zahl Priorität. In dem UNIM wird das Blocknummernregister als der SOF-Kanal (Paketanfang) angesehen. Bei anderen Ausführungsformen ist denkbar, sie an andere SOF-Kanäle zu senden.

ETE. External Transceiver Enable. Enabelt die Verwendung eines externen USB-Transceivers, im Gegensatz zu dem internen USB-Transceiver. Die Voreinstellung ist 0. Wenn sie nicht enabelt sind, werden alle Ausgaben der externen Transceiver-Schnittstelle bestätigt und alle Eingaben werden ignoriert. Dieses Bit kann nur modifiziert werden, wenn der Funktionszustand rückgesetzt ist (FSR.FS = 0).

SFM. Short Frame (Kurzblock) Enable. Enabelt die Verwendung kürzerer Blockgrößen. Dies wird exklusiv für die Diagnose und Prüfung verwendet, um die Zeit zum Emulieren eines gesamten Blockes zu reduzieren.

DIAG. Diagnostic Mode (Diagnosemodus). Enabelt den Diagnosemodus, in dem zusätzliche Diagnosefunktionen zugänglich sind. Dies umfaßt den Zugriff auf Diagnoseregister und die Möglichkeit in solche Register, wie das TCR (siehe unten), zu schreiben.

## Funktions-Zustandsregister (FSR)

Ein ein Wort breites Lese/Schreibregister. Es wird zum Steuern des Funktionszustands des Gerätes verwendet. Beim Reset wird dieses Register gelöscht. Das FSR-Registerformat ist unten angegeben:

15	2	1	0
res	FS1	FS0	

FS1, FS0 wird von der Software verwendet, um den Funktionszustand des Gerätes zu steuern.

## Funktionszustand

00: USB-Reset. Dies ist der USB-Reset-Zustand. Er wird bei einem Modulreset oder mittels Software bei Erfassung eines USB-Resets aktiviert. Bei Aktivierung werden alle Endpunktkanäle disabelt. DCR.DEF und DCR.AE sollten mittels Software bei Aktivierung dieses Zustandes gelöscht werden. Bei Verlassen dieses Zustandes sollte DCR.DEF gesetzt werden, so daß das Gerät auf die voreingestellte Adresse anspricht.

01: Resume. In diesem Zustand wird eine Wiederaufnahme-"K"-Signal erzeugt. Dieser Zustand wird abhängig von einer externen Weckanforderung von dem Gerät aktiviert. Der Knoten muß während wenigstens einer Millisekunde in diesem Zustand sein und darf nicht länger als fünfzehn Millisekunden in diesem Zustand bleiben.

10: Operational. Dies ist der normale Betriebszustand. In diesem Zustand ist der Knoten für den Betrieb auf dem

USB konfiguriert.

11: Suspend. Der Aussetzzustand wird bei Erfassung eines Aussetzereignisses während des Betriebszustandes aktiviert. Während des Aussetzzustandes arbeiten die Transceiver in ihrem Niederleistungs-Aussetzmodus, und es ist zulässig, Takte von dem USB-Modul zu nehmen, um zusätzlich Energie zu sparen. Alle Endpunktfeld- und interne Zustände bleiben konserviert. Bei Erfassung von Busaktivität werden die Takte wieder aufgenommen, und das Bit ISR.RSMD wird gesetzt. Davon abhängig kann Software den Wiedereintritt in den Betriebszustand bewirken. Bei Erfassung einer externen Weckanforderung durch das Gerät löst die Software die Fernwiederaufnahme aus und erzwingt den Eintritt in den Wiederaufnahmestand.

Man beachte: Bei Realisierungsformen mit der HUB-Funktion (Verteiler) kann der Übergang vom Aussetzen zum Wiederaufnehmen und vom Wiederaufnehmen zum Betrieb automatisch erfolgen. Bei der Schnittstellendefinition sollten Vorkehrungen für Signale getroffen werden, welche die Zeitbereiche überschreiten, um diese Übergänge zu erzeugen.

#### Transceiver-Konfigurationsregister (TCR)

Ein ein Wort breites Lese/Schreib-Register. Dieses Register steuert den Betrieb des Transceivers. Die unsymmetrischen Empfängerschwelwerte und die Treiberflankensteigungen können eingestellt werden. Dieses Register kann nur modifiziert werden, wenn DCR.DIAGE 1 ist. Bei HUB-Ausführungsformen steuert dieses Register alle Transceiver. Der Rücksetzwert ist {8'b00000000, 3'b100, 5'b10000}. Das TCR-Registerformat ist unten gezeigt:

15	8	7	5	4	0
res für LS xcvr		VADJ		CADJ	

VADJ Spannungsschwelwert-Einstellung. Steuert die unsymmetrischen Empfängerschwelwerte.

CADJ Stromeinstellung. Steuert den Steuerstrom für die Treiberflankensteigung.

#### Funktionsadreibregister (FAR)

Ein doppelwortbreites Lese/Schreibregister. Enthält das Adreß-Enablebit und die sieben Bit breite USB-Funktionsadresse, die während des normalen Betriebs verwendet wird, nachdem alle Setup-Operationen beendet sind. Bei einem USB-Reset wird dieses Register gelöscht. Bis zu vier Funktionsadressen können in diesem Register unterstützt werden – FA0 bis FA3. Das FAR-Registerformat ist unten gezeigt:

15	8	7	6	0
reserviert		AE	FA	

[Entwurfsverbesserung für Mehrfachendpunktgerät]

31	30	24	23	22	16	15	14	8	7	6	0
AE3	FA3		AE2	FA2		AE1	FA1		AE0	FA0	

FA[6 : 0] Funktionsadresse. Die 7 Bit breite Funktionsadresse wird von diesem Gerät verwendet.

AE Adreß-Enable. Enabelt die Verwendung der Funktionsadresse, die in FAR gesetzt ist, wenn das entsprechende Voreinstellungsadreibbit (DEF) nicht gesetzt ist.

#### Endpunktsteuerregister (ECR)

Ein ein Wort breites Lese/Schreibregister. ECD wird verwendet, um Befehle bei einem spezifizierten Endpunkt auszuführen. Bei einem Reset wird dieses Register gelöscht. Das ECR-Registerformat ist unten gezeigt:

15	8	7	4	3	0
res		EPN		CMD	

CMD Command. Dieser Befehl betrifft den Betrieb eines Endpunktes. Ein Wert 0 zeigt an, daß der Befehlsprozessor bereit für den nächsten Befehl ist.

0000: Complete. Gibt an, daß der vorhergehende Befehl abgeschlossen ist und der Endpunktcontroller bereit ist, einen weiteren Befehl anzunehmen.

0001: Input Enable. Bewirkt, daß der spezifizierte Endpunktkanal aus dem Stall- oder Disable-Zustand in den Eingang-Wartezustand geht. Der Steuerwortzeiger für den spezifizierten Endpunktkanal wird auf 0 zurückgesetzt. Wenn der aktuelle Zustand nicht Disable oder Stall ist, wird ein "F" zurückgegeben, um anzuzeigen, daß ein unzulässiger Befehl bei diesem Endpunkt versucht wurde, und der Befehl wird ignoriert. 5

0010: Input Ready. Bewirkt, daß der spezifizierte Endpunktkanal in den Eingang-Bereit-Zustand geht, wenn er momentan in dem Eingang-Warte-Zustand ist.

Dieser Befehl wird nach einem Steuerwort ausgegeben, wobei ein Eingang-Bereit-Pufferzustand in das Endpunktfeld für den entsprechenden Endpunkt geschrieben wird. 10

Dieser Befehl bewirkt nicht, daß das Steuerwort in dem Speicher gelesen wird, sondern er erlaubt vielmehr, daß es abhängig von dem Empfang eines IN-Tokens gelesen wird. Wenn der aktuelle Zustand des Endpunktes nicht der Wartezustand ist, wird der Befehl ignoriert. Wenn der aktuelle Zustand der Ausgabe- oder Setup-Zustand oder Disable oder Stall ist, wird ein "F" zurückgegeben, um anzuzeigen, daß bei diesem Endpunkt ein unzulässiger Befehl versucht wurde. 15

0011: Input Iso Ready. Bewirkt, daß der spezifizierte Endpunktkanal in den Eingang-ISO-Bereit-Zustand geht, wenn er momentan in dem Eingang-Wartezustand ist.

Dieser Befehl wird nach einem Steuerwort ausgegeben, wobei der Eingang-Bereit-Puffer-Zustand in das Endpunktfeld für den entsprechenden Endpunkt geschrieben wird.

Dieser Befehl bewirkt nicht, daß das Steuerwort in dem Speicher gelesen wird, sondern erlaubt vielmehr, daß es abhängig von dem Empfang eines IN-Tokens gelesen wird. Es wird erwartet, daß das Typfeld des Steuerwortes (TT) auf isochron gesetzt wird, dies wird jedoch nicht überprüft. Wenn der aktuelle Zustand des Endpunktes nicht Eingang-Warten ist, wird der Befehl ignoriert. Wenn der aktuelle Zustand einer der Ausgangs- oder Setupzustände oder Disabled oder Stall ist, wird "F" zurückgegeben, um anzuzeigen, daß bei diesem Endpunkt ein unzulässiger Befehl versucht wurde. 25

0100: Output Enable. Bewirkt, daß der spezifizierte Endpunktkanal aus dem Stall- oder Disabled-Zustand in den Ausgang-Wartezustand geht. Der Steuerwortzeiger für den spezifizierten Endpunkt wird auf 0 zurückgesetzt. Wenn der aktuelle Zustand nicht Disable oder Stall ist, wird ein "F" zurückgegeben, um anzuzeigen, daß ein unzulässiger Befehl bei diesem Endpunkt versucht wurde, und der Befehl wird ignoriert.

0101: Output Ready. Bewirkt, daß der spezifizierte Endpunktkanal in den Ausgang-Bereit-Zustand geht, wenn er momentan in dem Ausgang-Wartezustand ist. Dieser Befehl wird nach einem Steuerwort ausgegeben, wobei ein Ausgang-Bereit-Zustand in das Endpunktfeld für den entsprechenden Endpunkt geschrieben wird. 30

Dieser Befehl bewirkt nicht, daß das Steuerwort in dem Speicher gelesen wird, sondern erlaubt vielmehr, daß es abhängig von dem Empfang eines OUT-Tokens gelesen wird. Wenn der aktuelle Zustand des Endpunktes nicht Ausgang-Warte ist, wird der Befehl ignoriert. Wenn der aktuelle Zustand einer der Eingangszustände, Stall oder Disable ist, wird ein "F" zurückgegeben, um anzuzeigen, daß bei diesem Endpunkt ein unzulässiger Befehl versucht wurde, und dieser Befehl wird ignoriert. 35

0110: Output Iso Enable. Bewirkt, daß der spezifizierte Endpunktkanal aus dem Stall- oder Disable-Zustand in den Ausgang-Isochron-Wartezustand geht. Der Steuerwortzeiger für den spezifizierten Endpunkt wird auf 0 zurückgesetzt.

Wenn der aktuelle Zustand nicht Disable oder Stall ist, wird ein "F" zurückgegeben, um anzuzeigen, daß bei diesem Endpunkt ein unzulässiger Befehl versucht wurde, und der Befehl wird ignoriert. 40

0111: Output Iso Ready. Bewirkt, daß der spezifizierte Endpunktkanal in den Ausgang-Iso-Bereit-Zustand geht, wenn er momentan in dem Ausgang-Iso-Wartezustand ist. Dieser Befehl wird nach einem Steuerwort ausgegeben, wobei ein Ausgang-Bereit-Zustand in das Endpunktfeld für den entsprechenden Endpunkt geschrieben wird. 45

Dieser Befehl bewirkt nicht, daß das Steuerwort im Speicher gelesen wird, sondern er erlaubt vielmehr, daß es abhängig von dem Empfang eines OUT-Tokens gelesen wird. Wenn der aktuelle Zustand des Endpunktes nicht Ausgang-Iso-Warten ist, wird der Befehl ignoriert. Wenn der aktuelle Zustand einer der Eingangszustände, Stall oder Disable ist, wird ein "F" zurückgegeben, um anzuzeigen, daß bei diesem Endpunkt ein unzulässiger Befehl versucht wurde. 50

1000: Setup Ready. Bewirkt, daß der spezifizierte Endpunktkanal in den Setup-Bereit-Zustand geht. Dieser Befehl wird nach einem Steuerwort ausgegeben, wobei ein Ausgang-Bereit-Zustand in das Endpunktfeld für den entsprechenden Endpunkt geschrieben wird.

Dieser Befehl bewirkt nicht, daß das Steuerwort im Speicher gelesen wird, sondern er erlaubt vielmehr, daß es abhängig von dem Empfang eines SETUP-Tokens gelesen wird. Wenn der aktuelle Zustand des Endpunktes nicht Stall oder Disable ist, wird der Befehl ignoriert. 55

1001: Disable. Bewirkt, daß der spezifizierte Endpunktkanal in den Disable-Zustand geht. Dies erfolgt unmittelbar nach einem beliebigen anderen Zustand. Wenn der Endpunktkanal momentan aktiv ist, wird der Status bei der Steuerung geschrieben, wodurch angezeigt wird, daß der Endpunkt disabelt wurde.

1010: Stall. Bewirkt, daß der spezifizierte Endpunkt aus einem beliebigen Zustand in den Stall-Zustand geht. Dies erfolgt unmittelbar nach jedem beliebigen Zustand. Wenn der Endpunktkanal momentan aktiv ist, wird der Zustand in die Steuerung geschrieben, wodurch angezeigt wird, daß der Endpunkt angehalten wurde. 60

1011-1101: Reserviert. Bewirkt die Ausführung keiner Funktion. Gibt 0 zurück.

1110: Toggle CWP. Bewirkt, daß der Steuerwortzeiger umgeschaltet (getoggelt) wird. Dies wird in Verbindung mit Steuerübertragungen verwendet, wenn eine ungerade Anzahl Pakete während der Datenphase gesendet oder empfangen wird. 65

1111: Error. Gibt an, daß ein für den momentanen Zustand unzulässiger Befehl des aktuellen Endpunktkanals versucht wurde, und daß ein neuer Befehl angenommen werden könnte.

EPN Endpoint Pipe Number (Endpunktkanalnummer). Dieser Parameter bestimmt den Endpunktkanal, auf den der Befehl ausgeführt werden soll.

Vermerk für den Entwurf: Dieses Register sollte in der Hochgeschwindigkeitsbus-Schnittstellenlogik realisiert werden, wobei Daten über die asynchrone Schnittstelle weitergegeben werden. Zwei Signale werden von dem Endpunktcontroller zurückgegeben – eines, um anzuzeigen, daß der Befehl verarbeitet wurde und daß der Befehlswert auf "0" gelöscht werden sollte, und ein anderes, um anzuzeigen, daß der Befehl nicht verarbeitet werden konnte und auf "F" gesetzt werden sollte.

## Interrupt Status Register (ISR)

Ein ein Wort breites Lese/Schreibregister. Benachrichtigt die Software über Ereignisse, die aufgetreten sind, die einen Interrupt verursacht haben und weitere Verarbeitung erfordern. Bits werden gelöscht, indem 1 in das Bit/die Bits geschrieben wird, welche gelöscht werden sollen. Beim Reset wird dieses Register gelöscht. Das ISR-Registerformat ist unten gezeigt:

15	11	11	9	8	7	6	5	4	3	2	1	0
reserviert	HAS	ULD	FCI	SOFD	EOPD	RSMD	SD5	SD	RSTD	SECI	ECI	

ECI Endpoint Controller Interrupt (Endpunktcontroller-Interrupt). Es ist an einem der Endpunkte ein Ereignis aufgetreten, daß Softwareunterstützung benötigt. Dies ist das ODER aller EER.Epn, die mit EEM.Epn UND-verknüpft sind.

SECI Synchronized Endpoint Controller Interrupt (synchronisierter ECI). Eine verzögerte Version des ECI, der bei einem Inkrement oder dem Laden von FNR.FN abgetastet wird.

RSTD Reset Detected (Reset erfaßt). Ein Übergang auf SEO wurde erfaßt und während minimal 2,5 µs gehalten, während der Funktionszustand nicht USB Reset war. Als Reaktion sollte der Funktionszustand (DCR.FS) auf USBReset gesetzt werden und während wenigstens 100 µs USBReset bleiben. Nach dem Reset kann der Funktionszustand auf dem USB-Betriebszustand zurückgehen.

SD Suspend Detected (Aussetzen erfaßt). Ein Minimum von 3,0 µs Wartezustand wurde erfaßt, wodurch angezeigt wird, daß der Betrieb dieses Gerätes ausgesetzt werden sollte. Das Aussetzen (Suspend) erfolgt softwaregesteuert durch Schreiben von DCR.FS in den Aussetzwert.

SD5 Suspend Detected (Aussetzen erfaßt) 5. Ein Minimum von 5,0 µs Wartezustand wurde erfaßt, wodurch angezeigt wird, daß dieses Gerät nun eine Fern-Weckoperation ausführen darf. Sollten die Takte in dem Intervall, nachdem SD gesetzt, SD5 jedoch nicht gesetzt ist, disabled werden, dann sollte der verbleibende Teil der zusätzlichen 2 ms abgemessen werden, wenn die Takte wieder aufgenommen werden.

RSMD Resume Detected (Wiederaufnahme erfaßt). Ein Wiederaufnahmesignal wurde auf dem USB erfaßt, während dieses Gerät in dem Aussetzzustand (DCR.FS = Suspend) war, wodurch angezeigt wird, daß dieses Gerät seine Aufwecksequenz beginnen und in den Betriebszustand gehen sollte. Wenn die Takte disabled sind, wenn die Wiederaufnahme neu erfaßt wird, wird dieses Bit gesetzt, nachdem die Takte wieder enabled sind. Diese Wiederaufnahmebedingung sollte indem USB-Modul ohne Taktunterstützung zwischengespeichert und dann gelöscht werden, wenn das Wiederaufnahme-Erfaßt-Bit gesetzt ist.

EOPD End of Packet Detected. Eine gültige Packetendesequenz wurde auf dem USB erfaßt. Dies wird verwendet, wenn das Gerät eine Fernwiederaufnahme auslöst, um anzuzeigen, daß die Wiederaufnahmesequenz bestätigt und von dem Host beendet wurde.

SOFD Start of Frame Detected. Ein gültiges Blockanfangspaket wurde auf dem USB erfaßt.

FCI Frame Counter Interrupt. Der Blockzähler hat den programmierten Offset erreicht, d. h.  $TR.TI[7:0] = FNR.FN[7:0]$ .

ULD Unlocked Locked Detected. Gibt an, daß der Blockzeitgeber aus einem verriegelten in einen unverriegelten Zustand gegangen ist oder aus einem unverriegelten in den verriegelten Zustand zurückgegangen ist, was durch den aktuellen Wert von FNR.UL bestimmt wird, das Unlocked-Statusbit.

Man beachte, daß eine Einstellung der isochronen Warteschlange bei isochronen Endpunkten notwendig sein kann, weil FNR.FN[0] verwendet wird, um die nächste isochrone Übertragung zu wählen.

HST Holding Setup Token. Ein Setup-Token wurde empfangen, für den im voraus kein SRDY-Puffer zugewiesen wurde. Dieses Setup-Paket wird in dem Datenpfad gehalten, während darauf gewartet wird, daß dem Steuerendpunkt ein SRDY-Puffer zugewiesen wird.

Wenn dies bei mehr als einem Endpunkt möglich ist, muß der Endpunktzustand untersucht werden, um zu ermitteln, welcher Endpunkt auf den Puffer wartet. In diesem Fall sollte ein SRDY-Puffer auf CW0 gesetzt werden, ein OCOMP-Puffer wird auf CW1 gesetzt, und ein SRDY-Befehl wird an den zugehörigen Endpunktkanal übergeben. Man beachte, daß dieser Zustand vermieden wird, außer bei ungünstigen Bedingungen. Im Intervall zwischen dem Setzen dieses Bits und dem Zuweisen des Puffers werden alle gültigen IN- und OUT-Token mit NAK quittiert.

Man beachte: Eines der reservierten Bits ist für einen Hub-Interrupt reserviert, falls diese Funktion enthalten ist.

## Interrupt Maskenregister (IMR)

Ein ein Wort breites Lese/Schreibregister. IMR hat dasselbe Bitlayout wie das Interrupt-Statusregister. Die Bits wer-

# DE 199 00 369 A 1

den mit den entsprechenden Bits in dem Interrupt-Statusregister UND-verknüpft, das Ergebnis wird ODER-verknüpft, um das USB\_INT-Signal zu erzeugen. Bei einem Reset wird dieses Register gelöscht.

## Endpunkt Ereignisregister (EER)

Ein doppelwortbreites Lese/Schreibregister. EER wird verwendet, um anzuzeigen, auf welchem Endpunktkanal Ereignisse aufgetreten sind, die einen Dienst erfordern. Bei einem Reset wird dieses Register gelöscht. Das EER-Registerformat ist unten angegeben:

15	0
EE(0:15)	

{Konzept für zukünftige Verbesserungen}

31	16	15	0
reserviert		EE(0:15)	

EE(0 : 15) Endpoint Event. Ein Bit pro Endpunkt. Das Bit 0 wird dem Endpunkt 0 zugewiesen. Das Bit 1 dem Endpunkt 1, etc. Gibt an, daß ein Ereignis bei einem Endpunkt aufgetreten ist, daß Softwareunterstützung benötigen könnte. Die Bits werden gelöscht, indem 1 in die zu löschenden Bits geschrieben wird.

Dieses Bit wird gesetzt, nachdem der Status von dem Endpunktcontroller in dem Speicher auf CW.BS geschrieben wurde.

Wenn das CW.IC-Bit des Steuerwortes für diesen Endpunkt gesetzt ist, tritt das Ereignis auf, wenn CW.BS in eine der Fertig- oder Fehlerbedingungen geschrieben wird oder disabled oder unterbrochen ist. {8, 9, 10, 11, 12, 13, 14, 15}. Für isochrone Übertragungen wird der Ausgangs-Wartestatus auch als Beendigungsstatus angesehen {4, 5, 6}.

Wenn das CW.IE-Bit des Steuerwortes für diesen Endpunkt gesetzt ist, tritt das Ereignis auf, wenn CW.IS in einen der Fehlercodes geschrieben wird oder disabelt {10, 11, 12, 13, 14, 15}. Dies umfaßt nicht den CRC-Fehler und Bitstuffingfehler-Codes außer für isochrone Übertragungen.

Wenn sowohl das CW.IC- als auch das CW.IE-Bit des Steuerwortes für diesen Endpunkt gesetzt sind, tritt das Ereignis auf, wenn CW.IS in einen der Fehlercodes geschrieben wird oder disabled oder unterbrochen ist. Dies umfaßt den CRC-Fehler und Bitstuffingfehler-Codes {3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}.

Bemerkung für den Entwurf: Diese Ereignisbits sollten erst nach dem Aktualisieren des Speicherwortes im Speicher gesetzt werden, nicht nach dem Schreiben des internen CW.BS-Wertes. Dies dient dazu, zu verhindern, daß der Software-Interrupt angezeigt wird, bevor die Software den richtigen Status lesen kann.

## Endpunktmaskenregister (EMR)

Ein doppelwortbreites Lese/Schreibregister. Beim Reset wird dieses Register gelöscht. Das EMR-Registerformat ist unten angezeigt:

15	0
EM(0:15)	

{Entwurf für zukünftige Verbesserungen}:

31	16	15	0
reserviert		EM(0:15)	

EM(0 : 15) Endpoint Mask. Wird für jeden Endpunkt enabelt, um zu ermitteln, ob ein Interrupt erzeugt werden sollte, wenn das entsprechende Bit gesetzt ist. Das Bit 0 wird dem Endpunkt 0 zugeordnet, das Bit 1 dem Endpunkt 1, etc.

## Blocknummernregister (FNR)

Ein ein Wort breites Nur-Lese-Register. FNR sieht die aktuelle Blocknummer vor, die in dem SOF-Packet empfangen wird. Bei USB-Reset wird dieses Register auf C000h gesetzt. Das FNR-Registerformat ist unten gezeigt:



15	14	13	11	10	0
MF	UL	reserviert		FN	

FN Frame Number (Blocknummer). Die aktuelle Blocknummer, wie sie in dem letzten SOF-Paket empfangen wurde. Der Wert wird inkrementiert, wenn ein SOF verpaßt wird. Wenn zwei aufeinanderfolgende Blöcke fehlen oder fehlerhaft sind, wird die Blocknummer mit der nächsten Blocknummer aus einem gültigen SOF-Paket geladen. Das niedrigstwertige Bit der Blocknummer wird dazu verwendet, zu ermitteln, welcher isochrone Puffer als nächstes verwendet werden sollte.

UL Unlocked Flag. Das unlocked Flag (UL) wird gesetzt, um anzuzeigen, daß wenigstens zwei Blöcke ohne eine erwartete Blocknummer empfangen wurden und daß diese Blocknummer aus dem nächsten SOF-Paket in FN geladen wird. Bei einem Reset wird dieses Flag auf 1 gesetzt.

MF Missed SOF-Flag. Das verpaßte SOF-Flag (MF) wird gesetzt und zeigt eine Diskontinuität der Blocknummer an. Dieses Flag wird gesetzt, wenn die Blocknummer in einem gültigen empfangen SOF nicht mit dem erwarteten nächsten Wert übereinstimmt. Das Flag wird auch gesetzt, wenn innerhalb von 12060 Bitzeiten (FLMAX mit Toleranz) nach einer vorhergehenden Änderung von FNR.FN kein SOF empfangen wird. Bei einem Reset wird dieses Flag auf 1 gesetzt.

#### Blocknummer-Operation

Bei Empfang eines gültigen SOF-Pakets:

```

wenn ( (rcvdSOF.frame_number == FNR+1) oder (FNR.UL ist 1) )
    {FNR.FN <- rcvdSOF.frame_number, FNR.MF <- 0, FNR.UL <- 0}
sonst {
    FNR.FN <- FNR.FN + 1;
    wenn (FNR.MF ist 1 und FNR.UL ist 0) {FNR.UL <- 1; ISR.ULD <- 1}
    FNR.MF <- 1;
}

```

Ohne Empfang eines gültigen SOF-Pakets innerhalb von 12060 Bitzeiten (oder 796, wenn das Kurzblockbit DCR.SFM gesetzt ist) seit dem vorhergehenden Laden von FNR.FN

```

FNR.FN <- FNR.FN + 1;
Wenn ((FNR.MF ist 1 und FNR.UL ist 0) {FNR.UL <- 1; ISR.ULD <- 1} FNR.MF <- 1;

```

#### Referenzzählregister (RCR)

Ein ein-Wort-breites Nur-Leseregister. Wird zum Überwachen eines externen Bezugstaktes verwendet. Bei Reset wird dieses Register gelöscht. Das RCR-Registerformat ist unten gezeigt:

15	0
RC	

RC Reference Count (Bezugszählwert). Dies ist ein Zählwert der Anzahl der steigenden Flanken, die bei dem Bezugseingangssignal des Taktes des UNIM erfaßt werden (siehe TBD). Wenn es zusammen mit der Blocknummer gelesen wird, kann es eine präzise Erfassung der relativen Frequenz des externen Bezugstaktes im Verhältnis zum USB-Takt liefern. Die Differenz kann dem USB-Host über Feedback-Pakete angezeigt werden, um das SOF-Paket zu verzögern oder zu beschleunigen, damit der USB-Takt sich mit dem externen Takt synchronisiert. Um für die Software einen gemeinsamen Bezugspunkt aufrechtzuerhalten, wird der Referenzzählwert in dieses Register eingelesen, wenn FNR.FN mit einem neuen Wert geladen oder inkrementiert wird.

#### Zeitintervallregister (TIR)

Ein ein Wort breites Lese/Schreibregister. Enthält eine Anforderung eines Interrupts, wenn der Blockzähler den programmierten Offset das nächste Mal erreicht. Bei Reset wird dieses Register gelöscht. Das TIR-Registerformat ist unten gezeigt:

15	8	7	0
reserviert		TI	

TI Timer Interrupt Interval. Bewirkt, daß ISR.FCI das nächste Mal gesetzt wird, wenn der Blockzähler FNR.FN diesen Wert erreicht.

#### USB-Indexregister (UIR)

Ein ein Wort breites Lese/Schreibregister. Indexiert die Register, die in dem USB-Bereich liegen und auf die nicht häufig zugegriffen werden muß. Das UIR-Registerformat ist unten gezeigt:

15	10	9	8	7	0
res		SEL		Index	

SEL Selector. Wählt den Block der Register, auf die zugegriffen werden soll.

- 0: Endpunktkanal-Zustandsregister
- 1: Kanalabbildungsregister
- 2: Reserviert für USB-Hubregister
- 3: Diagnoseregister
- andere: reserviert

Index Index. Sieht den Blockindex für den ausgewählten Registerblock gemäß nachstehender Beschreibung vor.

#### USB-Datenregister (UDR)

Ein ein Wort breites Nur-Lese/Schreibregister. Lesezugriffe sind auf alle Stellen erlaubt. Das Datenregister wird mit den Daten geladen, die für das von dem UIR zu jeder Zeit ausgewählte Register gelesen werden. Das Schreiben in dieses Register ist in dem USB-Hubregister nur erlaubt, wenn UIR.SIL 2 ist. Ein Statusbit wird im Bit 15 definiert, um anzuzeigen, daß eine Schreiboperation erfolgreich war. Dies ist nur notwendig, wenn die Hubfunktion enthalten ist.

#### Endpunktkanal-Zustandsindex

Indexiert den Endpunktkanal-Zustandsvektor gemäß des unten gezeigten Indexierschemas:

7	4	3	0
res		EPN	

EPN Endpoint Pipe Number (Endpunktkanalnummer). Gibt die Endpunktkanalnummer an, für die der Endpunktkanal-Zustandsvektor in dem USB-Datenregister lesen kann.

#### Endpunktkanalzustand

Dies ist der 8 Bit-Endpunktkanalzustand, der für jeden Endpunktkanal gespeichert ist, und der für den momentan indexierten Endpunktkanal zurückgegeben wird. Wenn ein ungültiger Endpunktkanal indexiert wird, wird 0 zurückgegeben.

7	6	5	4	3	0
CTL	CWP	res		PES	

EPS Endpoint Pipe State (Endpunktkanalzustand). Der 4-Bit-Endpunktkanalzustand:

Zustand	Symbol	Wert
Disable	DIS	0
Eingang-Warten	I_IDL	1
Eingang-Bereit	I_RDY	2
Eingang-Aktiv	I_ACT	3
Angehalten	STL	4
Eingang hat ACK verpaßt	I_MAK	5
Eingang-ISO-Bereit	IS_RDY	6
Eingang-ISO-Aktiv	IS_ACT	7
Setup-Bereit	S_RDY	8
Ausgang-Warten	O_IDL	9
Ausgang-Bereit	O_RDY	10
Ausgang-Aktiv	O_ACT	11
Setup-Aktiv	S_ACT	12
Ausgang-ISO-Warten	OS_IDL	13
Ausgang-ISO-Bereit	OS_RDY	14
Ausgang-ISO-Aktiv	OS_ACT	15

Man beachte: Zu einer gegebenen Zeit ist nur ein Endpunktkanal aktiv.

CTL Control. Dies wird bestätigt, wenn der Endpunktkanal Steuerfolgen annehmen kann, welche den SETUP-Token betreffen. Dieses Bit wird gesetzt, nachdem für diesen Endpunktkanal CMD.Setup-Bereit vorlag, und bleibt gesetzt, bis der Endpunkt disabled wird.

CWP Control Word Pointer (Steuerwortzeiger). Wenn das Steuerwortbit 0 ist, wird auf das Steuerwort 0 dieses Endpunkts in dem Endpunktfeld als nächstes zugegriffen. Wenn das Steuerwortbit 1 ist, wird auf das Steuerwort 1 für diesen Endpunkt als nächstes zugegriffen. Es wird auf 0 gesetzt, wenn der Endpunkt enabelt wird.

#### Kanalabbildungsindex

Indexiert die Kanalabbildung, um die Abbildung auf den Endpunktkanal mit den folgenden Eingangsparametern zu bestimmen:

7	6	5	4	3	0
FAN		TT		EN	

FAN Function Address Number (Funktionsadrefnummer). Gibt bis zu einer von vier möglichen Funktionsadressen an. Bei allen Ausführungsformen ist die Funktionsadresse 0 immer vorhanden.

TT Token Type. Gibt den Tokentyp an:

00 OUT  
10 IN  
11 SETUP  
01 SOF

**EN Endpoint Number (Endpunktnummer).** Gibt die Endpunktnummer an, die auf dem USB empfangen wird –0 bis 15 einschließlich.

#### Kanalabbildungsdaten

Ein ein Byte breites Register. Gibt dem Endpunktkanal den aktuellen Wert des USB-Indexregisters zurück, das sich hierauf bezieht. Das Lesen und Schreiben dieses Registers ist nur gültig wenn DCR.ECE gleich 0 ist. In dieses Register kann nicht bei allen Ausführungsformen geschrieben werden. Bestimmte Werte der Endpunktnummer sind reserviert.

h FF zeigt an, daß der Wert von PMIR nicht auf einen Endpunktkanal abgebildet wird. h FE zeigt an, daß der Wert von PMIR auf den speziellen SOF-Kanal abgebildet wird, der das FNR-Register enthält.

7	0
EPPN	

**EPN Endpoint Pipe Number.** Gibt die Endpunktkanalnummer an, auf welche die Funktionsadresse, der Tokentyp und die Endpunktnummer abgebildet werden.

#### Hub-Register

Indexiert die USB-Hubregister gemäß dem folgenden Format:

7	5	4	0
res		HREG	

**HREG Hub Register.** Gibt die Hub-Register an, auf die über das USB-Datenregister zugegriffen werden soll. Für die tatsächlichen Registerdefinitionen dieser Register

Hub-Register	Name
0	Hub Control A
1	Hub Control B – DR
2	Hub Control C – PPCM
3	Hub Test Control A
4	Hub Status
6-15	reserviert
16	Port 1 Status
17	Port 1 Status Change
18	Port 2 Status
19	Port 2 Status Change
20	Port 3 Status
21	Port 3 Status Change
22	Port 4 Status
23	Port 4 Status Change
24	Port 5 Status
25	Port 5 Status Change
26	Port 6 Status
27	Port 6 Status Change
28	Port 7 Status
29	Port 7 Status Change
30-31	reserviert

Hub Data. Ein ein Byte breites Register. Erlaubt das Lesen und Schreiben der einzelnen Hubregister. Siehe das LM1050-Datenblatt für Einzelheiten dieser Register.

#### Diagnose-Register

Indexiert die USB-Diagnoseregister gemäß dem folgenden Format:

7	5	4	0
res	DREG		

DREG Diagnostic Register. Gibt die Diagnose-Register an, auf die durch das USB-Datenregister zugegriffen werden soll.

Diagnose-Register	Name	Reset-Wert	D7	D6	D5	D4	D3	D2	D1	D0
0	Port 0	00	VP	VM	RCV	res				
1	Hub Port 1		siehe LM1050 interne Spezifikation							
2	Hub Port 2									
3	Hub Port 3									
4	Hub Port 4									
5	Hub Port 5									
6	Hub Port 6									
7	Hub Port 7									
8	Frame Timer 0									
9	Frame Timer 1									
10	Frame Timer 2									
11	Frame Timer 3									
16	PHY A		reserviert					TXS		
17	MAC A		reserviert			MAC State				
18	MAC B		INV	SOF	ACK	D0	D1	SU	OU T	IN
19	MAC C		OVR	BTE	C5	C16	res	res	RXDPS	
20	MAC D		UDR	reserviert			TXDPS			

21-23	reserviert		reserviert	
24	EPCOT A		reserviert	cEPP
25	EPCOT B		HS	reserviert
26	EPCOT C		DP	BS
27	EPCOT D		reserviert	
28-30	reserviert		reserviert	
31	Revision		01	

#### Endpunktfeld-Basisadreßregister (EABAR)

Ein Doppelwort-breites Lese/Schreibregister. Basisadresse im Speicher des Blocks, der das Endpunktfeld enthält. Es muß zu einer 128-Bytegrenze ausgerichtet werden, d. h. die unteren sieben Bits werden auf 0 gesetzt (dies ist eine 25 Bit-Adresse). Bei Reset wird dieses Register gelöscht. Das EABAR-Registerformat ist unten gezeigt:

31	7	6	0
EABA		0000000	

Merke: Diese Basisadresse kann modifiziert werden, um schnell zwischen verschiedenen Konfigurationen umzuschalten. Nach dem Aufwecken aus einem Aussetzmodus während der Konfiguration kann zum Beispiel eine Konfiguration der Endpunkte wünschenswert sein, und nach der Konfiguration mehrerer verschiedener Konfigurationen kann es wünschenswert sein, verschiedene Kombinationen aus Schnittstellen und Endpunkten zu verwenden. Beim Umschalten der Konfigurationen sollten alle Endpunkte disabelt oder angehalten werden, und der Kontext aller Endpunkte wird dann durch einen Schreibvorgang in dieses Register verändert.

#### Pufferseite A-Register (BPA)

Ein Doppelwort-breites Lese/Schreibregister. Die oberen Bits der Pufferseite A. Ein 21 Bit-Register, das eine 2K-Byte-Seite definiert; in der Puffer für die Datenübertragung definiert sind. Diese Pufferseite wird von dem Pufferseitenauswahl-Bitwert in dem Steuerwort bestimmt, das in den Speicher des Endpunktfeldes geschrieben ist. Bei einem Reset wird dieses Register gelöscht. Das BPA-Registerformat ist unten gezeigt:

31	11	10	0
BPA		00000000000	

#### Pufferseite B-Register (BPB)

Ein Doppelwort-breites Lese/Schreibregister. Die oberen Bits der Pufferseite B. Ein 21-Bit-Register, das eine 2K-Byte-Seite definiert, in der Puffer für die Datenübertragung definiert sind. Diese Pufferseite wird von dem Pufferseitenauswahl-Bitwert von dem Steuerwort bestimmt, das in den Speicher in dem Endpunktfeld geschrieben ist. Bei einem Reset wird dieses Register gelöscht. Das BPB-Registerformat ist unten gezeigt:

31	11	10	0
BPB		00000000000	



## 2.1.9 Verwendungstips

Dieser Abschnitt sieht Anmerkungen vor, die zum Entwickeln von Treibern nützlich sind, welche das UNIM verwenden. Dies umfaßt die Initialisierung, Setup/Aufzählung und Datenübertragung für die Steuerung, Bulk-Endpunkte, Interrupt-Endpunkte und isochrone Endpunkte.

## Initialisierung

Nach einem Modul-Reset erscheint das Modul als ein nichtangeschlossenes Gerät auf dem USB, weil beide USB-Datenleitungen auf einen niedrigen Pegel gezogen sind. Zu dieser Zeit sind alle Endpunktkanäle disabelt, und der Funktionszustand ist zurückgesetzt.

Um ein Gerät als an einen stromaufwärtigen USB-Port angeschlossen zu erfassen, muß DCR.NAT gesetzt sein. Dadurch zieht das Gerät die USB-Datenleitung nicht mehr auf den niedrigen Pegel.

Wenn das Gerät einmal angeschlossen ist, muß der Endpunktkanalbetrieb initialisiert werden. Zunächst wird ermittelt, wieviele Endpunkte für eine Anwendung notwendig sind, sowie die Arten der Endpunkte und die Speichermenge, die für den Betrieb zugewiesen werden muß. Gestützt hierauf werden EABAR, BPA und BPB initialisiert. Das EABAR ist der Basiszeiger für das Feld, welche die Steuerworte enthält. Für jeden Endpunkt werden zwei Doppelworte zugewiesen. Dieser Zeiger muß in eine 128-Byte-Grenze fallen. Die BPA und BPB werden, wenn notwendig, zugewiesen, um den Basiszeiger für alle Datenübertragungen vorzusehen.

Da zu dieser Zeit noch nicht ganz feststehen kann, welche Endpunkt enablt werden sollen und wieviel Speicher diesen zugewiesen werden soll, müssen nur EABAR und BPA geladen werden, damit die erste Steuerübertragung verarbeitet werden kann.

Ein SRDY-Puffer wird in den CW0 des Endpunktkanals 0 gelegt und ein SRDY-Befehl wird ausgegeben, ECR.(SRDY,0).

Zu dieser Zeit werden die anderen Bits in dem DCR-Register modifiziert, einschließlich ETE, EPMS nach Bedarf, und ECE und DEF0 werden gesetzt.

Danach wird das Funktionszustandsregister in den Betriebszustand gesetzt (FSR.Operational).

Der empfangene Setup wird in den SRDY-Puffer geladen. Nachdem der Setup empfangen wurde, wird er analysiert, um zu ermitteln, wie reagiert werden muß. Wenn es sich um ein SET-ADDRESS-Setup-Paket handelt, wird die Funktionsadresse (FAR) mit der empfangenen Adresse geladen und das Enable-Bit wird gesetzt (FAR.AE). Das DEF-Bit wird mittels Hardware beim nächsten ACK zurückgesetzt, welches abhängig von einem IN-Token empfangen wird, oder bei dem nächsten Setup, der an diesem Endpunkt empfangen wird, falls das ACK verloren wurde.

## Interrupt Endpunkt Operation

Interrupt-Endpunkte sehen neue Interruptinformation für den Host vor, falls vorhanden. Wenn keine Interruptinformation anhängig ist, wird NAK zurückgegeben. Zum Initialisieren eines Interrupt-Endpunktkanals wird ein Eingang-Enable-Befehl an den richtigen Eingangskanal gegeben (ECR.(IENB, EPN)). Dies bewirkt, daß der Endpunktkanal in seinen Eingangs-Wartezustand geht, I\_IDL. Abhängig von dem Empfang eines IN-Tokens bei einem Endpunkt, der auf diesen Endpunktkanal abgebildet ist, wird NAK als Handshake zurückgegeben. Wenn Interruptdaten vorhanden sind, werden sie in einen Puffer geschrieben, das nächste Steuerwort an dem Endpunkt wird in I\_RDY geschrieben, und ein Befehl wird ausgegeben (ECR.(IRDY, EPN)), damit der Endpunktkanal in den IRDY-Zustand geht. Abhängig vom Empfang des nächsten IN-Tokens wird das Steuerwort bei EABAR + EPN + CWP gelesen, und dann werden die Daten übertragen. Wenn dem Toggle-Protokoll gefolgt wird, muß das Toggle-Enable-Bit des Steuerworts in dem Steuerwort gesetzt werden. Wenn dieser Endpunkt für ein Ratensteuerungs-Feedback verwendet wird, muß das Toggle-Enable-Bit nicht gesetzt werden.

Am Ende der Übertragung, wenn ACK vom Host empfangen wird, wird ein ICMP-Status in den Puffer geschrieben. Dies bestätigt dann, daß die Übertragung beendet ist. Wenn kein ACK empfangen wurde und Toggle enablt ist, wird ein Zwischenstatus ACK-verpaßt (IMAK) geschrieben. Nach dem Schreiben des Status wird der andere Endpunktpuffer gelesen. Falls der Status eine Beendigung anzeigt, beginnt der Eingang-Warte-Zustand. Wenn in der Zwischenzeit ein weiteres Ereignis aufgetreten ist und ein weitere IRDY-Puffer auf das andere Steuerwort gelegt wurde, wird es zu dieser Zeit gelesen, und der Endpunkt geht wieder in den I\_RDY-Zustand.

## Bulk-Endpunkt Operation

Bulk-Endpunkte arbeiten ähnlich wie die Interrupt-Endpunkte, abgesehen davon, daß üblicherweise größere Datenübertragungen betroffen sind, und daß sie in der IN- und OUT-Richtung verlaufen. Bulk-Endpunkte folgen immer dem Toggle-Protokoll, so daß in allen Steuerworten TE gesetzt ist.

Ein besonderer Bulk-Endpunkt wird entweder in der Eingangs- oder Ausgangsrichtung verwendet. Wenn ein Endpunkt das erste Mal initialisiert wird, wird ein Eingangs- oder Ausgangs-Enable-Befehl an die richtige Endpunktkanalnummer geschickt (ECR.(I/OENB, EPN)).

Bei Eingangsendpunkten werden Daten, wenn sie vorhanden sind in einen Puffer geschrieben, und ein Steuerwort wird mit einem IRDY-Zustand beschrieben, und TE wird gesetzt. Dann wird ein IRDY-Befehl ausgeführt. Maximal können zwei Datenpuffer zu einer gegebenen Zeit auf einem Endpunktkanal anhängig sein. Dies ermöglicht ein Doppelpuffer-Verfahren, falls nötig, um das Senden von NAKs während großer Datenübertragungen zu vermeiden. Die Software-Synchronisierung erfolgt durch Untersuchen des Pufferzustandes. Nur die vollständigen Pufferzustände können überschrieben werden.

Bei Ausgangsendpunkten muß ein Puffer vorbereitet werden, in den die Daten übertragen werden. Wenn dieser Puffer

vorhanden ist, wird ein Steuerwort mit einem Zeiger darauf in das richtige Steuerwort geschrieben. Das Toggle-Enable-Bit sollte hier ebenfalls gesetzt werden. Ein ORDY-Befehl wird an den Endpunktkanal übergeben. Nachdem die Daten für diesen Puffer als Teil des OUT-Tokens empfangen wurden, wird das Steuerwort im Speicher aktualisiert.

Wenn die Größe des Datenpakets nicht bekannt ist, sollte das Pufferrundungsbit des Steuerwortes gesetzt werden. Dies erlaubt, Pakete, die kleiner als die Puffergröße sind, zu empfangen, ohne daß ein Fehlerstatus geschrieben wird. Wenn das Pufferrundungsbit nicht gesetzt ist, werden empfangene Pakete, die den zugewiesenen Puffer nicht genau füllen, mit einem Datenfehler-Statusfehler geschrieben. In jedem Fall wird der Statusfehler geschrieben, wenn mehr Daten als erwartet für diesen Endpunkt empfangen werden.

Wenn das Auftreten eines Fehlers während des Empfangs dieser Daten erfaßt wird, wird ein Pufferzwischenzustand in den Speicher zurückgeschrieben, und beim nächsten OUT-Token wird der Puffer erneut verwendet. Dies bedeutet, daß die Software sich nur mit dem Vorsehen der Puffer befassen muß, welche die Daten empfangen sollen, und nicht mit den Verstrickungen der Protokolle für erneute Versuche.

#### Isochrone Endpunktoperation

Isochrone Endpunkte folgen demselben Schema wie Bulk-Endpunkte, sie folgen jedoch keinem Toggle-Protokoll. Vielmehr werden sie zu dem niedrigstwertigen Bit des Blocknummernregisters synchronisiert.

Bei isochronen Eingangsendpunkten werden Daten in eine Warteschlange gelegt, wobei dieselben IRDY-Puffer verwendet werden. Wenn ein IN-Token bei einem isochronen Endpunkt empfangen wird, wird das Datenpaket gesendet, wenn das niedrigstwertige Bit des Blocknummernregisters mit dem Steuerwortzeiger übereinstimmt. Sonst werden keine Daten gesendet. Dies erlaubt die Synchronisierung der Datenübertragung zu einer bestimmten Blocknummer. Zum Auslösen der Übertragung bei einer bestimmten Blocknummer kann das Blocknummernregister gelesen werden. Ein Interrupt kann angefordert werden (ISR.FCT), indem TIR auf den angeforderten Wert des FNR gesetzt wird, zu dem die Übereinstimmung vorliegen muß. Ein Abgleich erfolgt nur mit den unteren acht Bit, so daß dieses Ereignis bis zu 256 ms in eine Warteschlange gelegt werden kann, bevor es benötigt wird. Wenn dieser Interrupt verarbeitet wird, kann der passende Puffer auf dem Endpunktkanal in eine Warteschlange gelegt werden.

Bei isochronen Ausgangsendpunkten bestimmt das Toggle-Enable-Bit, wie das LSB des Blocknummernregisters verwendet wird. Wenn TE nicht gesetzt ist, werden alle Daten in den Speicher geschrieben, wenn jedoch FNR.LSB nicht gleich CWP ist, wird ein Synchronisierungsfehlerstatus (SYNE) als Pufferzustand geschrieben. Wenn TE gesetzt ist, werden Daten nur gespeichert, wenn FNR.LSB gleich CWP ist. Wenn ein isochroner Ausgangsstrom mit der Blocknummer synchronisiert wird, kann der erste Puffer auf TE gesetzt werden, so daß das Schreiben nur beginnt, wenn eine Übereinstimmung der Blocknummer vorliegt. Dann können alle nachfolgenden Blöcke erfassen, ob es einen Synchronisierungsfehler gibt.

Man beachte, daß das Blocknummernregister bis zu zwei Blöcke weiterzählt, selbst wenn der SOF vorübergehend verlorengeht. Sollte der SOF für mehr als zwei Blöcke verlorengehen, wird Unlocked angezeigt (ISR.UL), wobei zu dieser Zeit alle nachfolgenden isochronen Übertragungen mit Vorsicht behandelt oder beendet werden sollen, bis der Blocknummernzähler wieder synchron läuft.

#### Steuerendpunkt-Operation

In einem üblichen System ist der Endpunkt 0 immer ein Steuerendpunkt. Andere Endpunkte dürfen auch Steuerendpunkte werden. Diese Funktion wird unterstützt.

Bei einem Steuerendpunkt sollten das Steuerwort 1 und das Steuerwort 0 immer zusammengeschrieben werden. Die Statusphase und das SRDY für die nächste Steuerübertragung sollten zum Beispiel immer zusammengeschrieben werden. Erst nach der Erfassung eines OCMP in CWO aufgrund eines Empfangs eines neuen Setup sollten das Steuerwort und/oder CWO aktualisiert werden. Wenn eine ungerade Anzahl Datenpakete während der Datenphase einer Steuerwort-Lese- oder Schreiboperation gesendet werden, bevor die Statusphase gesendet wurde, kann es notwendig sein, den Steuerwortzeiger einzustellen, weil die Statusphase in das Steuerwort 1 gelegt werden muß, und der nächste SRDY-Puffer muß in das Steuerwort 0 gelegt werden. Um dies zu erreichen, ist es möglich, entweder einen SKIP-Puffer einzufügen oder einen CWP-Toggle-Befehl auszuführen, bevor der ORDY/IRDY-Befehl für den Statusphasen-Puffer ausgeführt wird.

Sollte ein SETUP-Token bei einem Steuerendpunkt erfaßt werden, für den momentan kein SRDY-Puffer zur Verfügung steht, wird das Möglichste getan, den Setup zu empfangen und in dem Datenpfad zu speichern. Wenn ein gültiger Setup empfangen wurde, wird das ISR.HST-Ereignis bestätigt. Abhängig von dem ISR.HST-Interrupt sollten ein SRDY-Puffer in das Steuerwort 0 und ein SRDY-Befehl geschrieben werden. Dies bewirkt, daß der anhängige Setup in den Speicher geschrieben wird. Der SRDY-Befehl bewirkt, daß der Endpunkt als ein Steuerendpunkt behandelt wird, und er bleibt gesetzt, bis ein Eingabe-Enable oder Ausgabe-Enable-Befehl ausgeführt wird. Der SRDY-Puffer sollte so schnell wie möglich vorgesehen werden, weil dann, wenn das IIST-Bit gesetzt ist, es den Datenpfad für mögliche zusätzliche Datenübertragungen blockiert. In der Zwischenzeit werden alle Token mit NAK quittiert, falls anwendbar. Die Implikationen dieses ISO-Verkehrs können beträchtlich sein und müssen beachtet werden, genauso wie der Fall eines Dienstverlustes.

Nach dem Empfang eines Setups wird er von der Firmware überprüft, um die richtige Antwort zu bestimmen. An diesem Punkt setzt die Firmware zusätzliche Eingangs- oder Ausgangspuffer nach Bedarf ein. Wenn der Setup ein von der Funktion nicht unterstützter Befehl ist, wird ein neuer SRDY-Puffer in das Steuerwort 0 gelegt, und der SRDY-Befehl wird ausgegeben. Dies bewirkt, daß alle Token an diesem Endpunkt eine STALL-Quittung zurückgeben, bis der nächste Setup empfangen wird. In der Zwischenzeit, während die Firmware den Setup analysiert, werden als Antwort auf Token NAKs gesendet.

# DE 199 00 369 A 1

## 2.1.10 Bemerkungen für die Umsetzung

### Registerschnittstelle

Man beachte, daß alle Register in dem Kernbus-Taktbereich realisiert sind. Register, auf die mit dem USB-Index zugegriffen wird, und Datenregister sind in dem USB-Taktbereich realisiert. Viele der Register erfordern asynchrone Handshakes, um die Taktbereiche zu überschreiten.

Während ein Endpunktkanal in einem seiner Bereitzustände ist, muß abhängig von dem Empfang eines IN-, OUT- und SETUP-Tokens eine Antwort innerhalb von 6,5 USB-Bitzeiten gegeben werden. Für einen IN-Token besteht diese Antwort aus einem STALL- oder NAK-Handshakepaket oder einem Datenpaket. Für OUT- und SETUP-Token muß sich der Endpunktkanal darauf vorbereiten, das Datenpaket zu empfangen. In allen Fällen wird das entsprechende Steuerwort gelesen und dazu verwendet, zu ermitteln, wie die Daten zu übertragen sind.

Die Operation, welche den Datenpfad am stärksten prüft, erfolgt bei Empfang eines IN-Tokens, während der ausgewählte Endpunkt in dem Eingabe-Bereit-Zustand ist. In diesem Fall muß das Steuerwort bereit sein, und Daten müssen bis zu der Zeit bei der MAC-Schnittstelle zur Verfügung stehen, bei der die PID fertig übertragen ist. Der Entwurf sollte für diesen Fall optimiert werden.

### Kapitel 3

#### Gerätespezifikationen

Tabelle 3-1

#### Ausgangssignale

Symbol	Figur	Beschreibung	Referenz	Min*	Max*
USB Ausgangssignale					
$t_{Uoh}$		USB-Ausgang-Haltezeit	Nach R.E> 12 MHz Takt <sup>a</sup>	0	
$t_{Uov}$		USB-Ausgang-Gültigzeit	Nach R.E> 12 MHz Takt <sup>a</sup>		15 ns

a. Der USB-Takt von 12 MHz ist ein interner Takt und nur als Referenz angegeben.

Tabelle 3-1

#### Eingangssignale

Symbol	Figur	Beschreibung	Referenz	Min*	Max*
USB Eingangssignale					

Fig. 3-1

USB-Aussetz-Zeitablauf

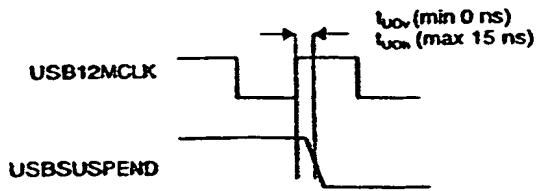


Fig. 3-2

Ablauf der USB-Sendesignale

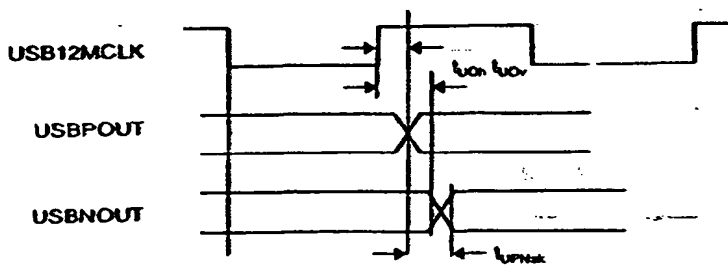


Fig. 3-3

USB OE-Zeitablauf

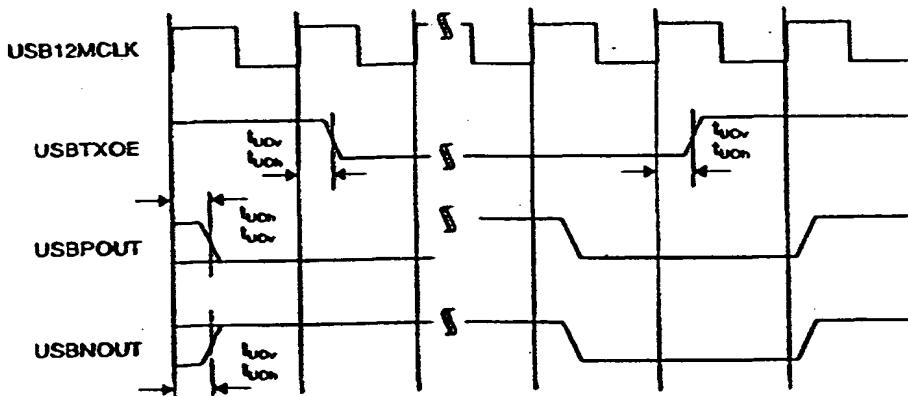
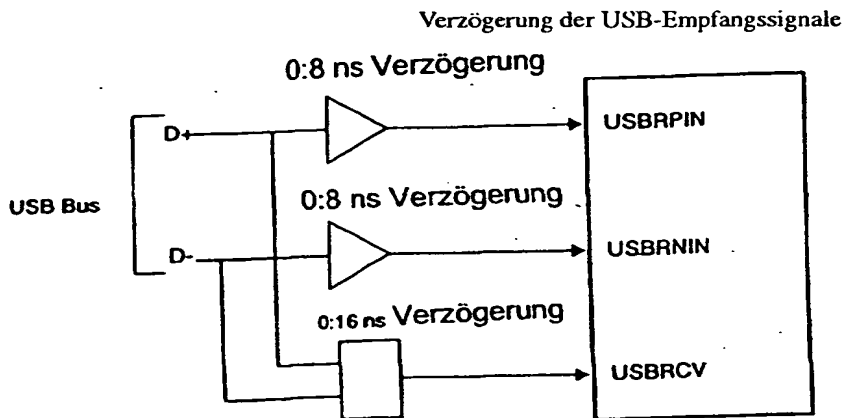


Fig. 3-4



**Beachte: Systemanforderungen für USB-Transceiver**

Beachte: USBXCU0 und USBXCU1, Wechselstrom und Gleichstrom erfüllen die Anforderung, die in Kapitel 7 der Universal Serial Bus Specification Version 1.0 spezifiziert sind.

#### Patentansprüche

1. Verfahren zum Betreiben eines Steuer-Endpunktes an einem Universeller Serialer Bus (USB), mit den folgenden Verfahrensschritten:

- Bereitstellen eines Sender-Empfängers, welcher zum Senden und Empfangen von Daten betreibbar mit dem USB verbunden ist;
- Bereitstellen eines Controllers, welcher mit dem Sender-Empfänger betreibbar verbunden ist, wobei der Controller Daten an den Sender-Empfänger zur Übertragung über den USB liefert, und wobei der Controller Daten annimmt, welcher der Sender-Empfänger über den USB empfängt;
- Bereitstellen eines Speichers, welcher funktionell mit dem Controller verbunden ist, wobei der Controller Daten in dem ersten Speicher sichern kann und Daten aus dem ersten Speicher kopieren kann;
- Bereitstellen einer Funktion, welche betreibbar mit dem Speicher und dem Controller verbunden ist, wobei die Funktion Befehle an den Controller liefern kann, Daten in dem Speicher sichern kann und Daten aus dem Speicher kopieren kann;
- Empfangen eines SETUP-Tokens von einem Host an dem USB, wobei der SETUP-Token anzeigt, daß der Host eine Steuerübertragung ausführen wird;
- Empfangen eines ersten DATA0-Paketes von dem Host auf dem USB, wobei das erste DATA0-Paket anzeigt, daß die Steuerübertragung eine Datenstufe mit wenigstens einer Datentransaktion umfassen wird; und Zuordnen eines Teiles des Speichers für die wenigstens eine Datentransaktion,

wobei die Funktion die Zuordnung ausführt und eine erste Speicheradresse an den Controller liefert, und wobei die erste Speicheradresse auf einen ersten Speicherstandort innerhalb des Teiles des Speichers zeigt.

2. Verfahren nach Anspruch 1, wobei das erste DATA0-Paket anzeigt, daß die Datentransaktionen IN-Transaktionen sind.

3. Verfahren nach Anspruch 2, wobei die Zuordnung eines Teiles des Speichers weiterhin die Zuordnung eines Teiles des Speichers umfaßt, der ausreichend ist, um die Daten für alle Transaktionen in der Datenstufe zu halten.

4. Verfahren nach Anspruch 3, wobei an der ersten Speicherstelle Daten für eine erste Datentransaktion gehalten werden.

5. Verfahren nach Anspruch 4, wobei die Zuordnung eines Teiles des Speichers weiterhin die Bereitstellung einer zweiten Speicheradresse an den Controller umfaßt, wobei die zweite Speicheradresse auf eine zweite Speicherstelle innerhalb des Teiles des Speichers zeigt.

6. Verfahren nach Anspruch 5, wobei der Controller Daten von der ersten Speicherstelle liest, und die Daten an den Sender liefert, um diese über den USB zu senden.

7. Verfahren nach Anspruch 6, wobei die Zuordnung eines Teiles des Speichers weiterhin die Bereitstellung einer dritten Speicheradresse an den Controller umfaßt, wobei die dritte Speicheradresse auf einen dritten Speicherstandort innerhalb des Teiles des Speichers zeigt, und wobei weiterhin die dritte Speicheradresse an den Controller nur geliefert wird, nachdem der Controller Daten von der ersten Speicherstelle kopiert hat.

8. Verfahren nach Anspruch 7, wobei auf der dritten Speicheradresse Daten für ein drittes Datenpaket gehalten werden.

9. Verfahren nach Anspruch 1, mit einem Schritt zum Empfangen eines unerwarteten SETUP-Tokens, und einem Speichern des unerwarteten SETUP-Tokens in einer Datenleitung.

10. Verfahren nach Anspruch 9, mit einem Schritt zur Benachrichtigung der Funktion über den unerwarteten SETUP-Token durch den Controller.

11. Verfahren nach Anspruch 10, mit einem Schritt zum Bereitstellen einer Speicheradresse an den Controller durch die Funktion, wobei die Speicheradresse auf einen Puffer im Speicher zeigt.

12. Vorrichtung zum Anschluß an einen Universal Serial Bus (USB), aufweisend:

- einen Pufferspeicher zur Speicherung von Daten;
- einen Controller, welcher funktionell mit dem Pufferspeicher und einem USB verbunden ist, der Controller aufweisend:

- 5       - einen ersten und einen zweiten Speicheradressen-Zeiger, wobei der erste und der zweite Speicheradressen-Zeiger eine Adresse innerhalb des Pufferspeichers halten;
- Logik für einen Endpunktkanal-Zustand zur Bewahrung des gegenwärtigen Zustandes eines Endpunktkanals, wobei die Logik anzeigt, ob der Endpunktkanal Daten zur Übertragung über den USB enthält;
- 10       - eine Funktion, welche funktionell mit dem Controller und dem Pufferspeicher verbunden ist, wobei die Funktion Daten in dem Pufferspeicher speichert und die entsprechende Adresse dem ersten und zweiten Pufferspeicher-Zeiger zuweist, wobei der Controller automatisch ein NAK sendet, wenn der Controller einen IN-Token einer Steuer-Lese-Übertragung empfängt, und wenn die Logik des Endpunktkanal-Zustandes anzeigt, daß der Endpunktkanal keine Daten enthält, und wobei der Controller automatisch die Daten sendet, wenn die Logik des Endpunktkanal-Zustandes anzeigt, daß der Endpunktkanal keine Daten enthält.

13. Vorrichtung nach Anspruch 12, wobei der Controller weiterhin eine Datenleitung für die temporäre Speicherung von Daten, die von dem USB empfangen wurden, aufweist, und wobei die Logik des Endpunktkanal-Zustandes weiterhin anzeigt, wenn der Endpunktkanal in einem Ready-Zustand zum Empfang einer SETUP-Transaktion ist, und wobei der SETUP-Token in der Datenleitung gespeichert wird, wenn ein SETUP-Token über den USB empfangen wird und die Logik des Endpunktkanal-Zustandes anzeigt, daß der Endpunktkanal nicht in einem Ready-Zustand ist.

14. Vorrichtung nach Anspruch 13, wobei der Controller weiterhin ein Interrupt-Bit aufweist, um anzuzeigen, daß der Kanalspeicher gegenwärtig einen SETUP-Token speichert.

15. Vorrichtung nach Anspruch 14, wobei die Funktion einen Puffer in dem Speicher zuordnet und den ersten Speicherzeiger einer Adresse in dem Puffer zuweist, wenn das Interrupt-Bit anzeigt, daß der Speicherkanal gegenwärtig einen SETUP-Token speichert.

16. Vorrichtung nach Anspruch 15, wobei die Funktion den ersten Speicherzeiger so zuweist, daß dieser auf einen Puffer zur Speicherung des SETUP-Tokens zeigt.

17. Vorrichtung nach Anspruch 13, wobei der Controller automatisch ein NAK sendet, während der SETUP-Token in der Datenleitung gehalten wird.

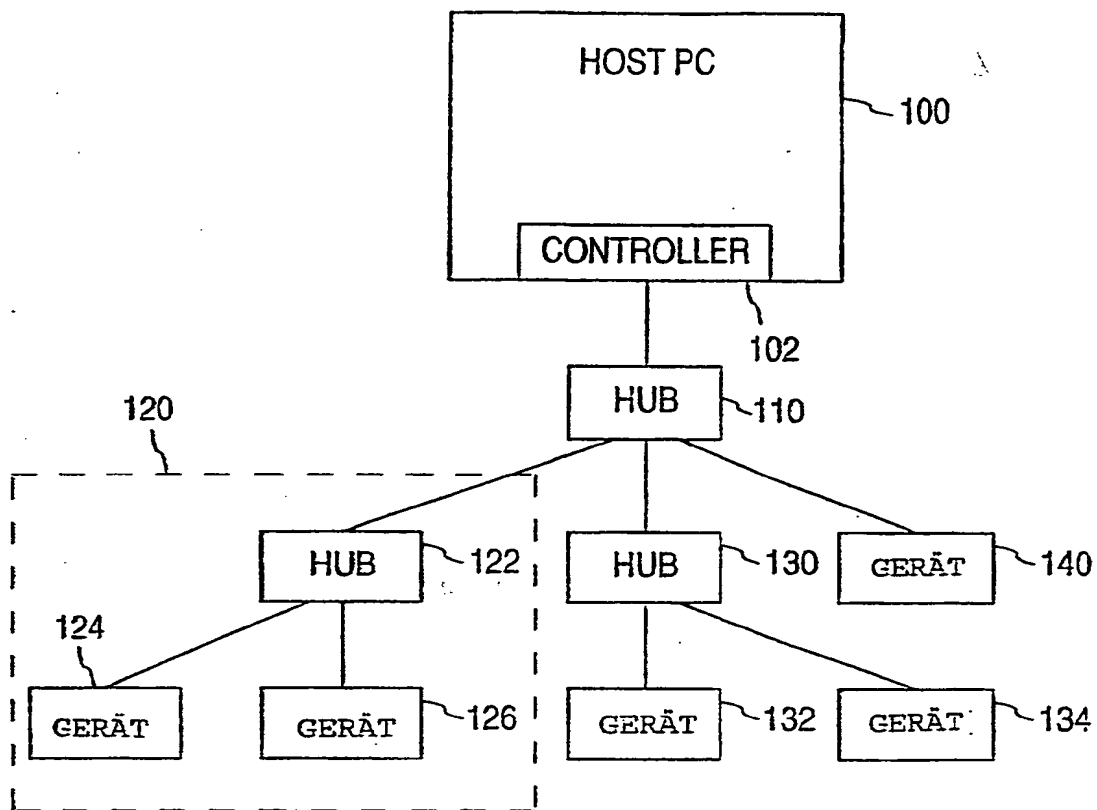
18. Vorrichtung nach Anspruch 17, wobei der Controller automatisch das in der Leitung gespeicherte Setup-Paket auf einen Speicherpuffer überträgt, und die Abarbeitung folgender Transaktionen fortsetzt.

---

Hierzu 7 Seite(n) Zeichnungen

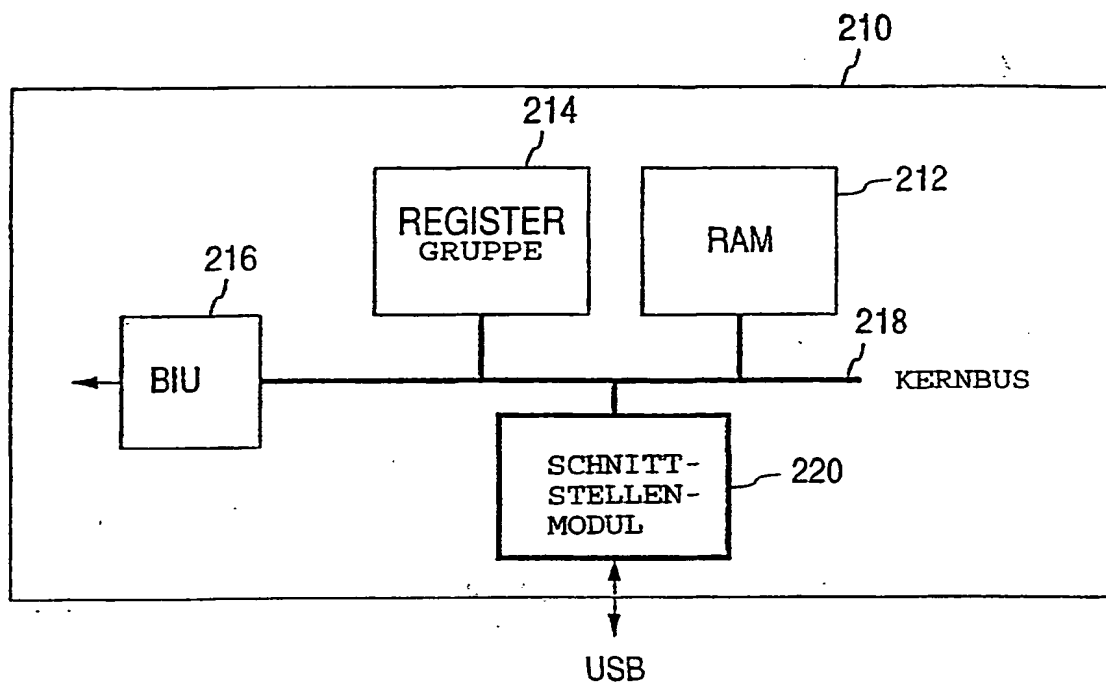
---

- Leerseite -

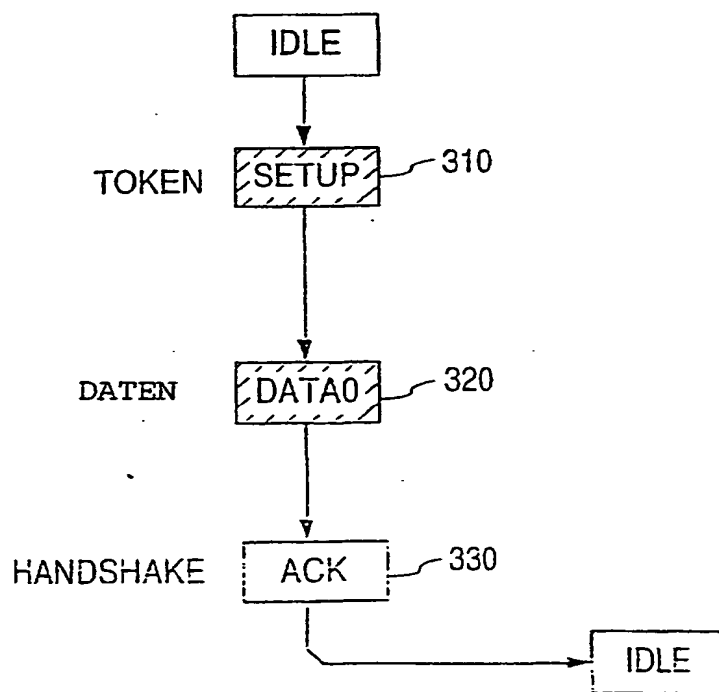


FIGUR 1

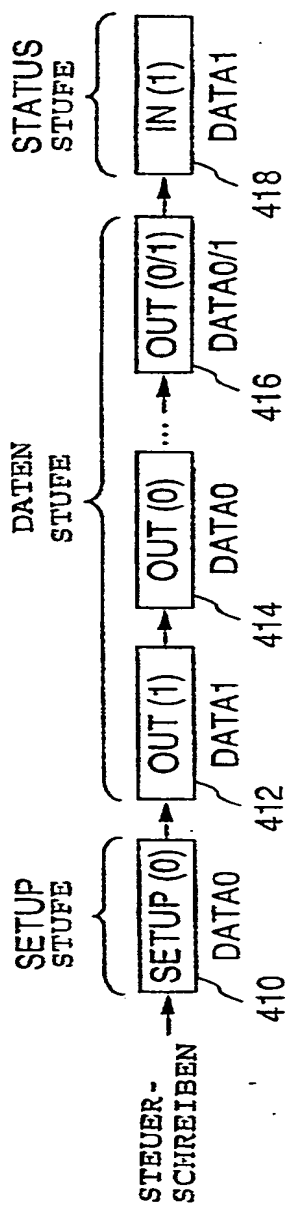




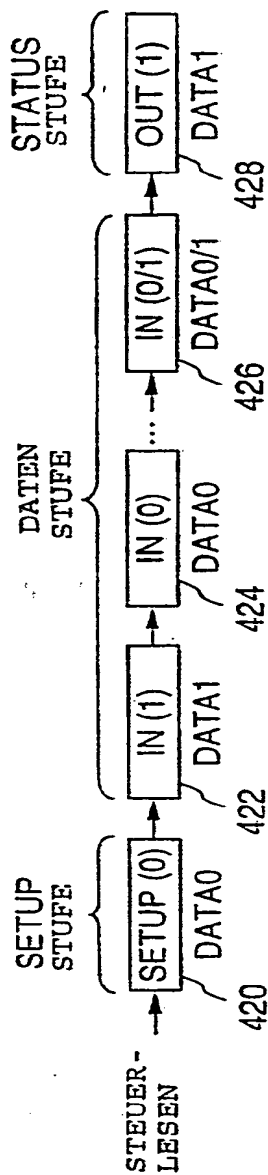
FIGUR 2



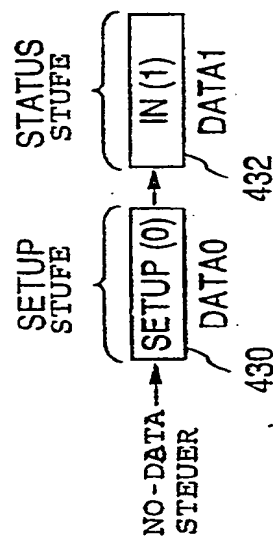
FIGUR 3



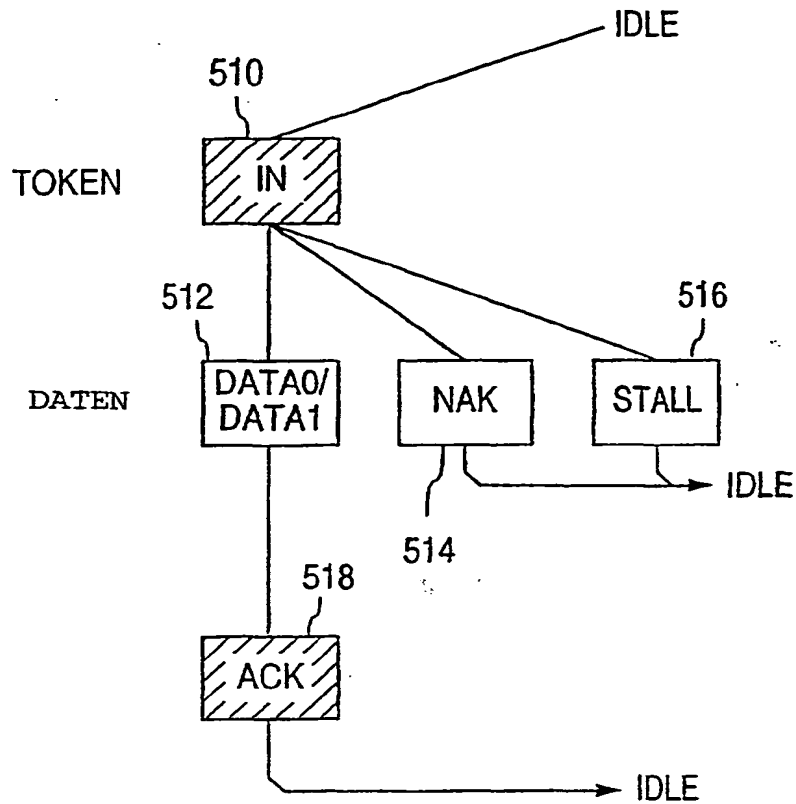
FIGUR 4A



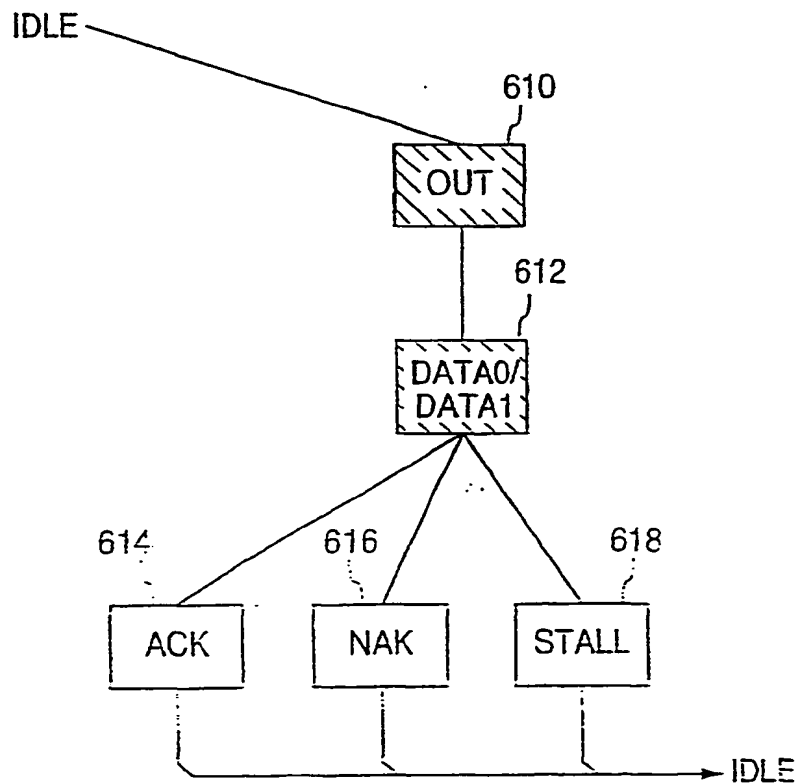
FIGUR 4B



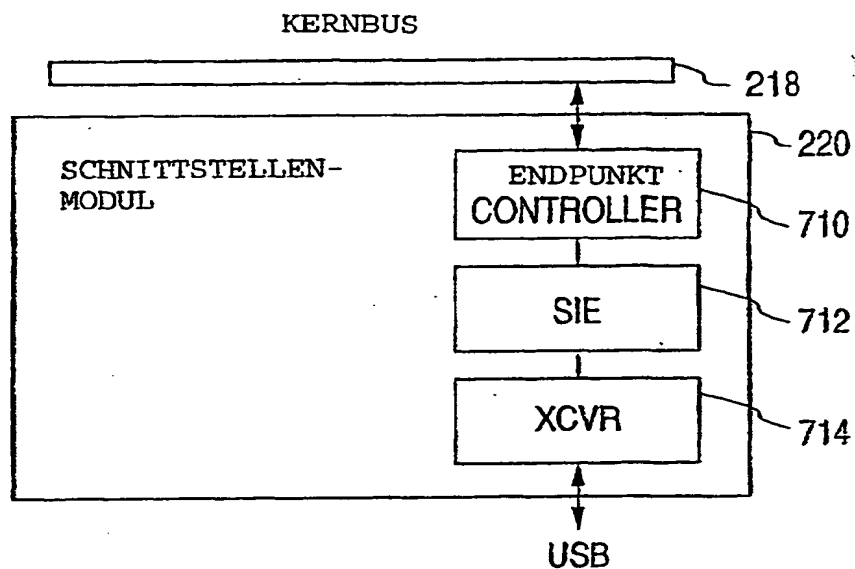
FIGUR 4C



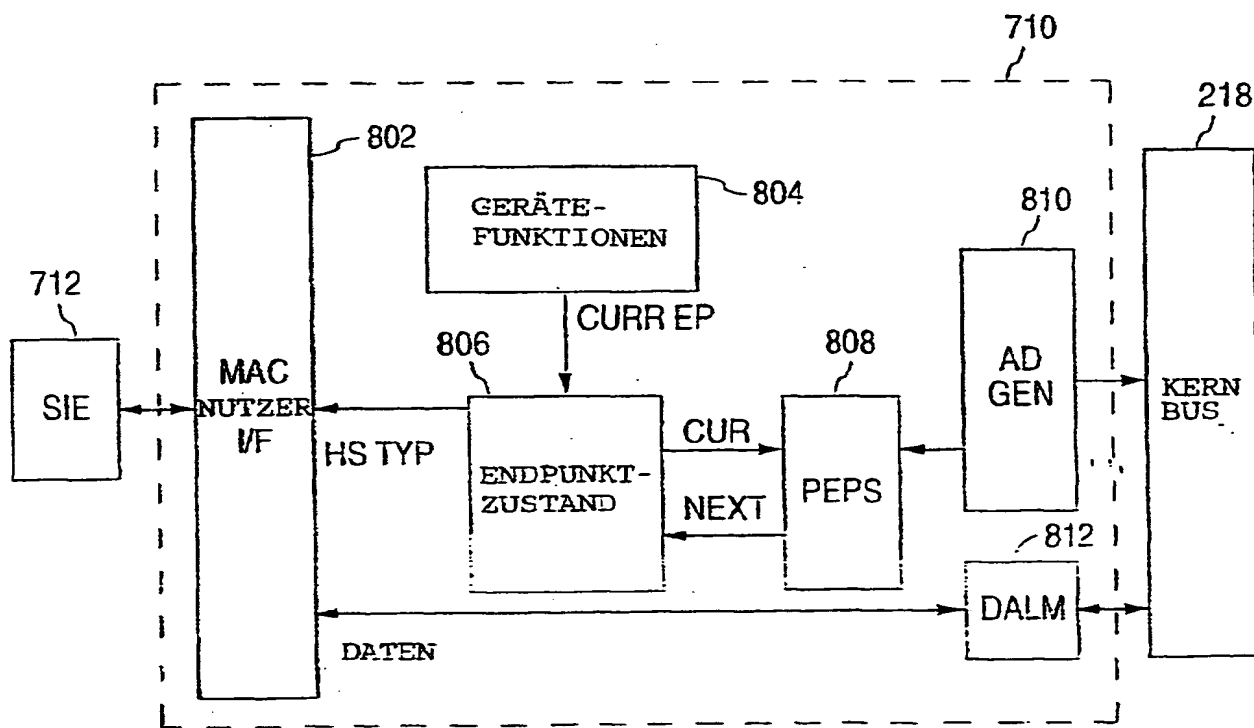
FIGUR 5



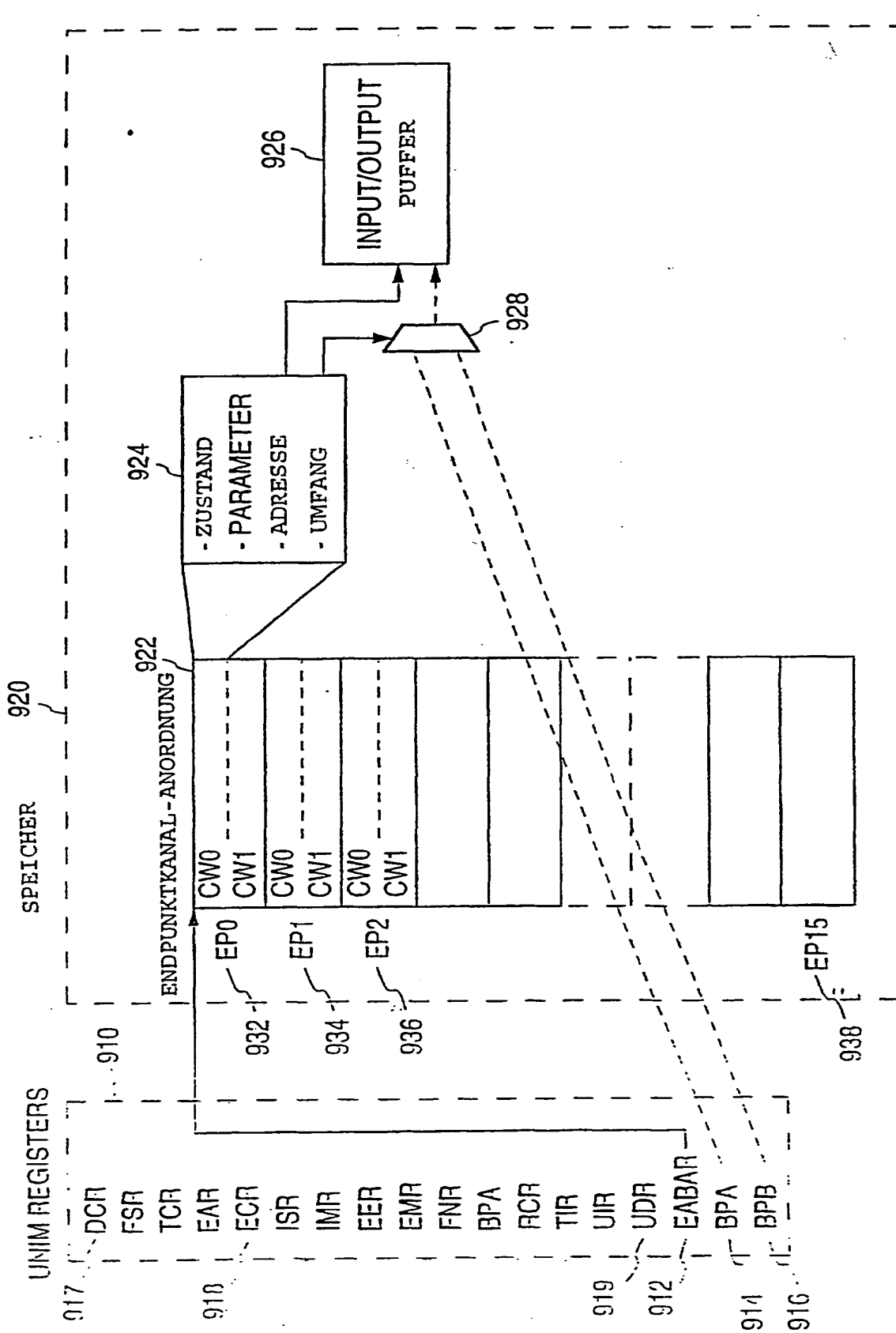
FIGUR 6



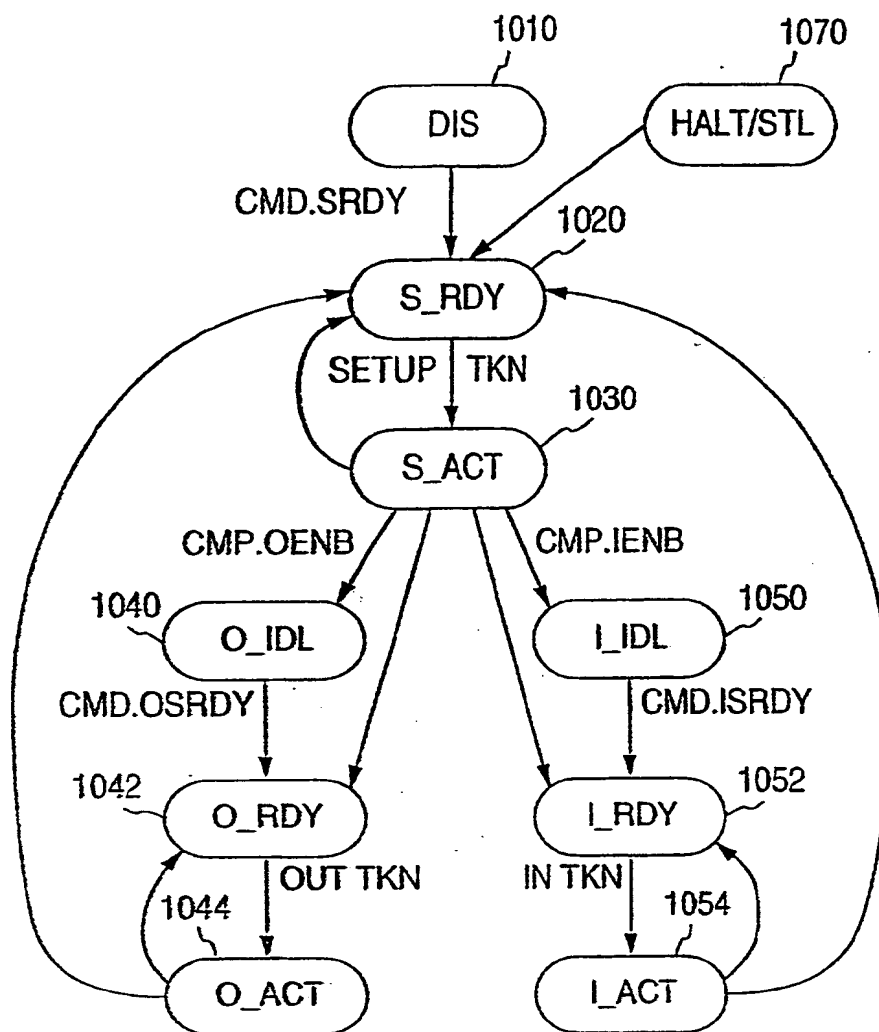
FIGUR 7



FIGUR 8



FIGUR 9



FIGUR 10